Computer-Aided Derivation of Multi-scale Models: A Rewriting Framework [*]

Bin Yang

Department of Applied Mathematics
Northwestern Polytechnical University
710129 Xi'an Shaanxi, China

University of Franche-Comté,
26 chemin de l'Epitaphe, 25030 Besançon Cedex, France
`bin.yang@femto-st.fr`

Walid Belkhir

INRIA Nancy - Grand Est,
CASSIS project,
54600 Villers-lès-Nancy, France

`walid.belkhir@inria.fr`

Michel Lenczner

FEMTO-ST, Département Temps-Fréquence,
University of Franche-Comté,
26 chemin de l'Epitaphe, 25030 Besançon Cedex, France
`michel.lenczner@utbm.fr`

February 12, 2013

Abstract

We introduce a framework for computer-aided derivation of multi-scale models. It relies on a combination of an asymptotic method used in the field of partial differential equations with term rewriting techniques coming from computer science. In our approach, a multi-scale model derivation is characterized by the features taken into account in the asymptotic analysis. Its formulation consists in a derivation of a reference model associated to an elementary nominal model, and in a set of transformations to apply to this proof until it takes into account the wanted features. In addition to the reference model proof, the framework includes first order rewriting principles designed for asymptotic model derivations, and second order rewriting principles dedicated to transformations of model derivations. We apply the method to generate a family of homogenized models for second order elliptic equations with periodic coefficients that could be posed in multi-dimensional domains, with possibly multi-domains and/or thin domains.

1 Introduction

There is a vast literature on multi-scale methods for partial differential equations both in applied mathematics and in many modeling areas. Among all developed methods, asymptotic methods occupy a special place because they have rigorous mathematical foundations and can lead to error estimates based on the small parameters involved in the approach. This is a valuable aspect from the model reliability point of view. They have been applied when a physical problem depends on one or more small parameters which can be some coefficients or can be related to the geometry. Their principle is to identify the asymptotic

model obtained when the parameters tend to zero. For instance, this method applies in periodic homogenization, i.e. to systems consisting of a large number of periodic cells, the small parameter being the ratio of the cell size over the size of the complete system, see for instance [BLP78, CD99, JZKO94]. Another well-developed case is when parts of a system are thin, e.g. thin plates as in [Cia], that is to say that some of their dimensions are small compared to others. A third kind of use is that of strongly heterogeneous systems e.g. [BB02], i.e. when equation coefficients are much smaller in some parts of a system than in others. These three cases can be combined in many ways leading to a broad variety of configurations and models. In addition, it is possible to take into account several nested scales and the asymptotic characteristics can be different at each scale: thin structures to a scale, periodic structures to another, etc.... It is also possible to cover cases where the asymptotic phenomena happen only in certain regions or even are localized to the boundary. Moreover, different physical phenomena can be taken into account: heat transfer, solid deformations, fluid flow, fluid-structure interaction or electromagnetics. In each model, the coefficients can be random or deterministic. Finally, different operating regimes can be considered as the static or the dynamic regimes, or the associated spectral problems. Today, there exists a vast literature covering an impressive variety of configurations.

Asymptotic methods, considered as model reduction techniques, are very useful for complex system simulation and are of great interest in the software design community. They enjoy a number of advantages. The resulting models are generally much faster (often by several order of magnitude – depending on the kind of model simplification –) to simulate than the original one and are fully parameterized. In addition, they do not require any long numerical calculation for building them, so they can be inserted into identification and optimization loops of a design process. Finally, they are of general use and they can be rigorously applied whenever a model depends on one or several small parameters and the error between their solution and nominal model solution can be estimated.

Despite these advantages, we observe that the asymptotic modeling techniques have almost not been transferred in general industrial simulation software while numerical techniques, as for instance the Finite Element Method, have been perfectly integrated in many design tools. The main limitation factor for their dissemination is that each new problem requires new long hand-made calculations that may be based on a relatively large variety of techniques. In the literature, each published paper focus on a special case regarding geometry or physics, and no work is oriented in an effort to deal with a more general picture. Moreover, even if a large number of models combining various features have already been derived, the set of already addressed problems represents only a tiny fraction of those that could be derived from all possible feature combinations using existing techniques.

Coming to this conclusion, we believe that what prevents the use of asymptotic methods by non-specialists can be formulated as a scientific problem that deserves to be posed. It is precisely the issue that we discuss in this paper. We would like to establish a mathematical framework for combining asymptotic methods of different nature and thus for producing a wide variety of models. This would allow the derivation of complex asymptotic models are made by computers. In this paper, we present first elements of a solution by combining some principles of asymptotic model derivations and *rewriting* methods issued from computer science.

In computer science equational reasoning is usually described by rewrite rules, see [BN98] for a classical reference. A rewrite rule $t \rightarrow u$ states that every occurrence of an instance of t in a term can be replaced with the corresponding instance of u. Doing so, a proof based on a sequence of equality transformations is reduced to a series of rewrite rule applications. Rules can have further conditions and can be combined by specifying strategies which specify where and when to apply them, see for instance [Ter03, CK01, CFK05, BKKR01, CKLW03]

The method developed in this paper is led by the idea of derivating models by generalization. For this purpose, it introduces a *reference model* with its derivation and a way to generate generalizations to cover

more cases. The level of detail in the representation of mathematical objects should be carefully chosen. On the one hand it should have enough precision to cover fairly wide range of models and on the other hand calculations should be reasonably sized. The way the generalizations are made is important so that they could be formulated in a single framework.

In this paper, we select as *reference problem* that of the periodic homogenization of a scalar second order elliptic equation posed in a one-dimension domain and with Dirichlet boundary conditions. Its derivation is based on the use of the two-scale transform operator introduced in [ADH90], and reused in [BLM96]. We quote that homogenization of various problems using this transformation was performed according to different techniques in [Len97, Len06, LS07, CD00, CDG02, CDG08]. Here, we follow that of [LS07], so a number of basic *properties* coming from this paper are stated and considered as the building blocks of the proofs. The complete derivation of the model is organized into seven *lemmas* and whose proof is performed by a sequence of applications of these *properties*. Their generalization to another problem requires generalization of certain properties, which is assumed to be made independently. It may also require changes in the path of the proof, and even adding new lemmas. The mathematical concepts are common in the field of partial differential equations: geometric domains, variables defined on these domains, functions of several variables, operators (e.g. derivatives, integrals, two-scale transform, etc..). Finally, the proofs of Lemmas are designed to be realizable by rewriting.

Then, we presents a computational framework based on the theory of rewriting to express the above method. Each *property* is expressed as a rewrite rule that can be conditional, so that it can be applied or not according to a given logical formula. A step in a lemma proof is realized by a strategy that expresses how the rule applies. The complete proof of a lemma is then a sequence of such strategies. Ones we use have been developed in previous work [BGL] that is implemented in Maple®, here we provide its formalization. To allow the successful application of rewriting strategies to an expression that contains associative and/or commutative operations, such as $+, *, \cup, \cap$, etc, we use the concept of rewriting modulo an equational theory [BN98, §11]. Without such concept one needs to duplicate the rewriting rules.

In this work, rewriting operates on expressions whose level of abstraction accurately reflects the mathematical framework. Concrete descriptions of geometric domains, functions or operators are not provided. Their description follows a grammar that has been defined in order that they carry enough information allowing for the design of the rewriting rules and the strategies. In some conditions of rewriting rules, the set of variables on which an expression depends is required. This is for example the case for the linearity property of the integral. Rather than introducing a typing system, which would be cumbersome and restrictive, we introduced a specific functionality in the form of a λ-term (i.e. a program). The language of strategy allows this use. Put together all these concepts can express a lemma proof as a strategy, i.e. a first order strategy, and therefore provide a framework of symbolic computation. The concept of generalization of a proof is introduced as second order rewrite strategies, made with second order rewriting rules, operating on first order strategies. They can transform first order rewrite rules and strategies and, where appropriate, remove or add new ones. This framework has been implemented in the software Maple®. We present its application to the complete proof of the reference problem and also to the generalizations of the first lemma, by applying second order strategies, to multi-dimensional geometrical domains, multi-dimensional thin domains and multi-domains.

The paper is organized as follows. Section 2 is devoted to all mathematical aspects. This includes all definitions and properties, the lemmas and their proof. The principles of rewrite rules and strategies are formulated in Section 3. Section 4 is devoted to the theoretical framework that allows to derive a model and its generalizations. Implementation results are described in section 6.

2 Skeleton of two-scale modeling

We recall the framework of the two-scale convergence as presented in [LS07], and the proof of the *reference model* whose implementation and extension under the form of algorithms of symbolic computation are discussed in Section 6. The presentation is divided into three subsections. The first one is devoted to basic definitions and properties, stated as *Propositions*. The latter are admitted without proof because they are assumed to be prerequisites, or building blocks, in the proofs. They are used as elementary steps in the two other sections detailing the proof of the convergence of the two-scale transform of a derivative, and the homogenized model derivation. The main statements of these two subsections are also stated as *Propositions* and their proofs are split into numbered blocks called lemmas. Each lemma is decomposed into steps refering to the definitions and propositions. All components of the *reference model* derivation, namely the definitions, the propositions, the lemmas and the proof steps are designed so that to be easily implemented and also to be generalized for more complex models. We quote that a number of elementary properties are used in the proof but are not explicitely stated nor cited.

2.1 Notations, Definitions and Propositions

Note that the functional framework used in this section is not as precise as it should be for a usual mathematical work. The reason is that the functional analysis is not covered by our symbolic computation. So, precise mathematical statements and justifications are not in the focus of this work.

In the sequel, $A \subset \mathbb{R}^n$ is a bounded open set, with measure $|A|$, having a "sufficiently" regular boundary ∂A and with unit outward normal denoted by $n_{\partial A}$. We shall use the set $L^1(A)$ of integrable functions and the set $L^p(A)$, for any $p > 0$, of functions f such that $f^p \in L^1(A)$, with norm $||v||_{L^p(A)} = (\int_A |v|^p \, dx)^{1/p}$. The Sobolev space $H^1(A)$ is the set of functions $f \in L^2(A)$ whose gradient $\nabla f \in L^2(A)^n$. The set of p times differentiable functions on A is denoted by $\mathcal{C}^p(A)$, where p can be any integer or ∞. Its subset $\mathcal{C}_0^p(A)$ is composed of functions whose partial derivatives are vanishing on the boundary ∂A of A until the order p. For any integers p and q, $\mathcal{C}^q(A) \subset L^p(A)$. When $A = (0, a_1) \times ... \times (0, a_n)$ is a cuboid (or rectangular parallelepiped) we say that a function v defined in $\mathbb{R}^n$ is A-periodic if for any $\ell \in \mathbb{Z}^n$, $v(y + \sum_{i=1}^n \ell_i a_i e_i) = v(y)$ where e_i is the ith vector of the canonical basis of $\mathbb{R}^n$. The set of A-periodic functions which are $\mathcal{C}^\infty$ is denoted by $\mathcal{C}_\sharp^\infty(A)$ and those which are in $H^1(A)$ is denoted by $H_\sharp^1(A)$. The operator tr (we say *trace*) can be defined as the restriction operator from functions defined on the closure of A to functions defined on its boundary ∂A. Finally, we say that a sequence $(u^\varepsilon)_{\varepsilon > 0} \in L^2(A)$ converges strongly in $L^2(A)$ towards $u^0 \in L^2(A)$ when ε tends to zero if $\lim_{\varepsilon \to 0} ||u^\varepsilon - u^0||_{L^2(A)} = 0$. The convergence is said to be weak if $\lim_{\varepsilon \to 0} \int_A (u^\varepsilon - u^0) v \, dx = 0$ for all $v \in L^2(A)$. We write $u^\varepsilon = u^0 + O_s(\varepsilon)$ (respectively $O_w(\varepsilon)$), where $O_s(\varepsilon)$ (respectively $O_w(\varepsilon)$) represents a sequence tending to zero strongly (respectively weakly) in $L^2(A)$. Moreover, the simple notation $O(\varepsilon)$ refers to a sequence of numbers which simply tends to zero. We do not detail the related usual computation rules.

Proposition 1 *[Interpretation of a weak equality] For $u \in L^2(A)$ and for any $v \in \mathcal{C}_0^\infty(A)$,*

$$if \int_A u(x) \, v(x) \, dx = 0 \ then \ u = 0$$

in the sense of $L^2(A)$ functions.

Proposition 2 *[Interpretation of a periodic boundary condition] For $u \in H^1(A)$ and for any $v \in \mathcal{C}_\#^\infty(A)$,*

$$if \int_{\partial A} u(x) \, v(x) \, n_{\partial A}(x) \, dx = 0 \ then \ u \in H_\sharp^1(A) \,.$$

In the remainder of this section, only the dimension $n = 1$ is considered, the general definitions being used for the generalizations discussed in Section 6.

Notation 3 *[Physical and microscopic Domains] We consider an interval $\Omega = \bigcup_{c=1}^{N(\varepsilon)} \Omega_c^{1,\varepsilon} \subset \mathbb{R}$ divided into $N(\varepsilon)$ periodic cells (or intervals) $\Omega_c^{1,\varepsilon}$, of size $\varepsilon > 0$, indexed by c, and with center x_c. The translation and magnification $(\Omega_c^{1,\varepsilon} - x_c)/\varepsilon$ is called the unit cell and is denoted by Ω^1. The variables in Ω and in Ω^1 are denoted by x^ε and x^1.*

The two-scale transform T is an operator mapping functions defined in the physical domain Ω to functions defined in the two-scale domain $\Omega^\sharp \times \Omega^1$ where for the *reference model* $\Omega^\sharp = \Omega$. In the following, we shall denote by Γ, $\Gamma^\sharp$ and Γ^1 the boundaries of Ω, $\Omega^\sharp$ and Ω^1.

Definition 4 *[Two-Scale Transform] The two-scale transform T is the linear operator defined by*

$$(Tu)(x_c, x^1) = u(x_c + \varepsilon x^1) \tag{1}$$

and then by extension $T(u)(x^\sharp, x^1) = u(x_c + \varepsilon x^1)$ for all $x^\sharp \in \Omega_c^{1,\varepsilon}$ and each c in $1, .., N(\varepsilon)$.

Notation 5 *[Measure of Domains] $\kappa^0 = \frac{1}{|\Omega|}$ and $\kappa^1 = \frac{1}{|\Omega^\sharp \times \Omega^1|}$.*

The operator T enjoys the following properties.

Proposition 6 *[Product Rule] For two functions u, v defined in Ω,*

$$T(uv) = (Tu)(Tv). \tag{2}$$

Proposition 7 *[Derivative Rule] If u and its derivative are defined in Ω then*

$$T\left(\frac{du}{dx}\right) = \frac{1}{\varepsilon}\frac{\partial (Tu)}{\partial x^1}. \tag{3}$$

Proposition 8 *[Integral Rule] If a function $u \in L^1(\Omega)$ then $Tu \in L^1(\Omega^\sharp \times \Omega^1)$ and*

$$\kappa^0 \int_\Omega u \ dx = \kappa^1 \int_{\Omega^\sharp \times \Omega^1} (Tu) \ dx^\sharp dx^1. \tag{4}$$

The next two properties are corollaries of the previous ones.

Proposition 9 *[Inner Product Rule] For two functions u, $v \in L^2(\Omega)$,*

$$\kappa^0 \int_\Omega u \ v \ dx = \kappa^1 \int_{\Omega^\sharp \times \Omega^1} (Tu) \ (Tv) \ dx^\sharp dx^1. \tag{5}$$

Proposition 10 *[Norm Rule] For a function $u \in L^2(\Omega)$,*

$$\kappa^0 \|u\|_{L^2(\Omega)}^2 = \kappa^1 \|Tu\|_{L^2(\Omega^\sharp \times \Omega^1)}^2. \tag{6}$$

Definition 11 *[Two-Scale Convergence] A sequence $u^\varepsilon \in L^2(\Omega)$ is said to be two-scale strongly (respect. weakly) convergent in $L^2(\Omega^\sharp \times \Omega^1)$ to a limit $u^0(x^\sharp, x^1)$ if Tu^ε is strongly (respect. weakly) convergent towards u^0 in $L^2(\Omega^\sharp \times \Omega^1)$.*

Definition 12 [Adjoint or Dual of T] *As T is a linear operator from $L^2(\Omega)$ to $L^2(\Omega^\sharp \times \Omega^1)$, its adjoint T^* is a linear operator from $L^2(\Omega^\sharp \times \Omega^1)$ to $L^2(\Omega)$ defined by*

$$\kappa^0 \int_\Omega T^*v \ u \ dx = \kappa^1 \int_{\Omega^\sharp \times \Omega^1} v \ Tu \ dx^\sharp dx^1. \tag{7}$$

The expression of T^* can be explicited, it maps regular functions in $\Omega^\sharp \times \Omega^1$ to piecewise-constant functions in Ω. The next definition introduce an operator used as a smooth approximation of T^*.

Definition 13 [Regularization of T^*] *The operator B is the linear continuous operator defined from $L^2(\Omega^\sharp \times \Omega^1)$ to $L^2(\Omega)$ by*

$$Bv = v(x, \frac{x}{\varepsilon}). \tag{8}$$

The nullity condition of a function $v(x^\sharp, x^1)$ on the boundary $\partial\Omega^\sharp \times \Omega^1$ is transferred to the range Bv as follows.

Proposition 14 [Boundary Conditions of Bv] *If $v \in \mathcal{C}_0^\infty(\Omega^\sharp; \mathcal{C}^\infty(\Omega^1))$ then $Bv \in \mathcal{C}_0^\infty(\Omega)$.*

Proposition 15 [Derivation Rule for B] *If v and its partial derivatives are defined on $\Omega^\sharp \times \Omega^1$ then*

$$\frac{d(Bv)}{dx} = B(\frac{\partial v}{\partial x^\sharp}) + \varepsilon^{-1} B(\frac{\partial v}{\partial x^1}). \tag{9}$$

The next proposition states that the operator B is actually an approximation of the operator T^* for Ω^1-periodic functions.

Proposition 16 [Approximation between T^* and B] *If $v(x^\sharp, x^1)$ is continuous, continuously differentiable in $x^\sharp$ and Ω^1-periodic in x^1 then*

$$T^*v = Bv - \varepsilon B(x^1 \frac{\partial v}{\partial x^\sharp}) + \varepsilon O_s(\varepsilon). \tag{10}$$

Conversely,

$$Bv = T^*(v) + \varepsilon T^*(x^1 \frac{\partial v}{\partial x^\sharp}) + \varepsilon O_s(\varepsilon). \tag{11}$$

Next, the formula of integration by parts is stated in a form compatible with the Green formula used in some extensions. The boundary Γ is composed of the two end points of the interval Ω, and the unit outward normal n_Γ defined on Γ is equal to -1 and $+1$ at the left- and right-endpoints respectively.

Proposition 17 [Green Rule] *If $u, v \in H^1(\Omega)$ then the traces of u and v on Γ are well defined and*

$$\int_\Omega u \frac{dv}{dx} \ dx = \int_\Gamma tr(u) \ tr(v) \ n_\Gamma \ ds(x) - \int_\Omega v \frac{du}{dx} \ dx. \tag{12}$$

The last proposition is stated as a building block of the homogenized model derivation.

Proposition 18 [The linear operator associated to the Microscopic problem] *For $\mu \in \mathbb{R}$, there exist $\theta^\mu \in H_\sharp^1(\Omega^1)$ solutions to the linear weak formulation*

$$\int_{\Omega^1} a^0 \frac{\partial\theta^\mu}{\partial x^1} \frac{\partial w}{\partial x^1} \ dx^1 = -\mu \int_{\Omega^1} a^0 \frac{\partial w}{\partial x^1} \ dx^1 \ for \ all \ w \in \mathcal{C}_\sharp^\infty(\Omega^1), \tag{13}$$

and $\frac{\partial\theta^\mu}{\partial x^1}$ is unique. Since the mapping $\mu \mapsto \frac{\partial\theta^\mu}{\partial x^1}$ from $\mathbb{R}$ to $L^2(\Omega^1)$ is linear then

$$\frac{\partial\theta^\mu}{\partial x^1} = \mu \frac{\partial\theta^1}{\partial x^1}. \tag{14}$$

Moreover, this relation can be extended to any $\mu \in L^2(\Omega^\sharp)$.

6

2.2 Two-Scale Approximation of a Derivative

Here we detail the *reference computation* of the weak two-scale limit $\eta = \lim_{\varepsilon \to 0} T(\frac{du^\varepsilon}{dx})$ in $L^2(\Omega^\sharp \times \Omega^1)$ when

$$\|u^\varepsilon\|_{L^2(\Omega)} \text{ and } \left\|\frac{du^\varepsilon}{dx}\right\|_{L^2(\Omega)} \leq C, \tag{15}$$

C being a constant independent of ε. To simplify the proof, we further assume that there exist u^0, $u^1 \in L^2(\Omega^\sharp \times \Omega^1)$ such that

$$T(u^\varepsilon) = u^0 + \varepsilon u^1 + \varepsilon O_w(\varepsilon),$$

i.e.

$$\int_{\Omega^\sharp \times \Omega^1} (T(u^\varepsilon) - u^0 - \varepsilon u^1)v \, dx^\sharp dx^1 = \varepsilon O(\varepsilon) \text{ for all } v \in L^2(\Omega^\sharp \times \Omega^1). \tag{16}$$

We quote that Assumption (16) is not necessary, it is introduced to simplify the proof since it avoids some non-equational steps. The statement proved in the remaining of the subsection is the following.

Proposition 19 [Two-scale Limit of a Derivative] *If u^ε is a sequence bounded as in (15) and satisfying (16), then u^0 is independent of x^1,*

$$\tilde{u}^1 = u^1 - x^1 \partial_{x^\sharp} u^0 \tag{17}$$

defined in $\Omega^\sharp \times \Omega^1$ is Ω^1-periodic and

$$\eta = \frac{\partial u^0}{\partial x^\sharp} + \frac{\partial \tilde{u}^1}{\partial x^1}. \tag{18}$$

Moreover, if $u^\varepsilon = 0$ on Γ then $u^0 = 0$ on $\Gamma^\sharp$.

The proof is split into four Lemmas corresponding to the first four blocks discussed in Section 6, the other three being detailed in subsection 2.3.

Lemma 20 [First Block: Constraint on u^0] *u^0 is independent of x^1.*

Proof. We introduce

$$\Psi = \varepsilon \kappa^0 \int_\Omega \frac{du^\varepsilon}{dx} Bv \, dx$$

with $v \in \mathcal{C}_0^\infty(\Omega^\sharp; \mathcal{C}_0^\infty(\Omega^1))$. From the Cauchy-Schwartz inequality and (15), $\lim_{\varepsilon \to 0} \Psi = 0$.

- **Step 1.** The Green formula (12) and Proposition 14 $\Longrightarrow$

$$\Psi = -\varepsilon \kappa^0 \int_\Omega u^\varepsilon \frac{d(Bv)}{dx} \, dx.$$

- **Step 2.** Proposition 15 $\Longrightarrow$

$$\Psi = \kappa^0 \int_\Omega u^\varepsilon B(\frac{\partial v}{\partial x^1}) \, dx + O(\varepsilon).$$

- **Step 3.** Proposition 16 $\Longrightarrow$

$$\Psi = \kappa^0 \int_\Omega u^\varepsilon T^*(\frac{\partial v}{\partial x^1}) \, dx + O(\varepsilon).$$

- **Step 4.** Definition 12 $\Longrightarrow$

$$\Psi = \kappa^1 \int_{\Omega^\sharp \times \Omega^1} T(u^\varepsilon) \frac{\partial v}{\partial x^1} \, dx + O(\varepsilon).$$

- **Step 5.** Assumption (16) and passing to the limit when $\varepsilon \to 0 \implies$

$$\kappa^1 \int_{\Omega^\sharp \times \Omega^1} u^0 \frac{\partial v}{\partial x^1} \, dx = 0.$$

- **Step 6.** The Green formula (12) and $v = 0$ on $\Omega^\sharp \times \Gamma^1 \implies$

$$\kappa^1 \int_{\Omega^\sharp \times \Omega^1} \frac{\partial u^0}{\partial x^1} \, v \, dx = 0.$$

- **Step 7.** Proposition 1 $\implies$

$$\frac{\partial u^0}{\partial x^1} = 0.$$

∎

Lemma 21 *[Second Block: Two-Scale Limit of the Derivative]* $\eta = \frac{\partial u^1}{\partial x^1}$.

Proof. We choose $v \in \mathcal{C}_0^\infty(\Omega^\sharp; \mathcal{C}_0^\infty(\Omega^1))$ in

$$\Psi = \kappa^1 \int_{\Omega^\sharp \times \Omega^1} T(\frac{du^\varepsilon}{dx}) v \, dx^\sharp dx^1. \tag{19}$$

- **Step 1.** Definition 12 $\implies$

$$\Psi = \kappa^0 \int_\Omega \frac{du^\varepsilon}{dx} T^* v \, dx.$$

- **Step 2.** Proposition 16 (to approximate T^* by B), the Green formula (12), the linearity of integrals, and again Proposition 16 (to approximate B by T^*) $\implies$

$$\Psi = -\kappa^0 \int_\Omega u^\varepsilon T^*(\frac{\partial v}{\partial x^\sharp}) \, dx - \frac{\kappa^0}{\varepsilon} \int_\Omega u^\varepsilon T^*(\frac{\partial v}{\partial x^1}) \, dx - \kappa^0 \int_\Omega u^\varepsilon T^*(\frac{\partial^2 v}{\partial x^1 \partial x^\sharp} x^1) \, dx + O(\varepsilon).$$

- **Step 3.** Definition 12 $\implies$

$$\begin{aligned}
\Psi &= -\kappa^1 \int_{\Omega^\sharp \times \Omega^1} T(u^\varepsilon) \frac{\partial v}{\partial x^\sharp} \, dx^\sharp dx^1 - \frac{\kappa^1}{\varepsilon} \int_{\Omega^\sharp \times \Omega^1} T(u^\varepsilon) \frac{\partial v}{\partial x^1} \, dx^\sharp dx^1 \\
&\quad - \kappa^1 \int_{\Omega^\sharp \times \Omega^1} T(u^\varepsilon) x^1 \frac{\partial^2 v}{\partial x^1 \partial x^\sharp} \, dx^\sharp dx^1 + O(\varepsilon).
\end{aligned}$$

- **Step 4.** Assumption (16) $\implies$

$$\begin{aligned}
\Psi &= -\kappa^1 \int_{\Omega^\sharp \times \Omega^1} u^0 \frac{\partial v}{\partial x^\sharp} \, dx^\sharp dx^1 - \frac{\kappa^1}{\varepsilon} \int_{\Omega^\sharp \times \Omega^1} u^0 \frac{\partial v}{\partial x^1} \, dx^\sharp dx^1 - \kappa^1 \int_{\Omega^\sharp \times \Omega^1} u^1 \frac{\partial v}{\partial x^1} \, dx^\sharp dx^1 \\
&\quad - \kappa^1 \int_{\Omega^\sharp \times \Omega^1} u^0 \frac{\partial^2 v}{\partial x^1 \partial x^\sharp} x^1 + O(\varepsilon).
\end{aligned}$$

- **Step 5.** The Green formula (12), Lemma 20, and passing to the limit when $\varepsilon \to 0 \implies$

$$\kappa^1 \int_{\Omega^\sharp \times \Omega^1} \eta \, v \, dx^\sharp dx^1 = \kappa^1 \int_{\Omega^\sharp \times \Omega^1} \frac{\partial u^1}{\partial x^1} v \, dx^\sharp dx^1.$$

- **Step 6.** Proposition 1 $\implies$

$$\eta = \frac{\partial u^1}{\partial x^1}.$$

■

Lemma 22 *[**Third Block: Microscopic Boundary Condition**] $\tilde{u}^1$ is Ω^1-periodic.*

   Proof. In (19), we choose $v \in \mathcal{C}_0^\infty(\Omega^\sharp; \mathcal{C}_\sharp^\infty(\Omega^1))$.

- **Step 1.** The steps 1-5 of the second block $\Longrightarrow$

$$\kappa^1 \int_{\Omega^\sharp \times \Omega^1} \eta v \ dx^\sharp dx^1 - \kappa^1 \int_{\Omega^\sharp \times \Gamma^1} (u^1 - x^1 \frac{\partial u^0}{\partial x^\sharp}) v \ n_{\Gamma^1} \ dx^\sharp dx^1 - \kappa^1 \int_{\Omega^\sharp \times \Omega^1} \frac{\partial u^1}{\partial x^1} v \ dx^\sharp dx^1 = 0.$$

- **Step 2.** Lemma 21 $\Longrightarrow$

$$\int_{\Omega^\sharp \times \Gamma^1} (u^1 - x^1 \frac{\partial u^0}{\partial x^\sharp}) v \ n_{\Gamma^1} \ dx^\sharp ds(x^1) = 0. \tag{20}$$

- **Step 3.** Definition (17) of $\tilde{u}^1$ and Proposition 2 $\Longrightarrow$

$$\tilde{u}^1 \text{ is } \Omega^1\text{-periodic.} \tag{21}$$

■

Lemma 23 *[**Fourth Block: Macroscopic Boundary Condition**] u^0 vanishes on $\Gamma^\sharp$.*

   Proof. We choose $v \in \mathcal{C}_0^\infty(\Omega^\sharp)$,

- **Step 1.** The steps 1-5 of the second block and $u^\varepsilon = 0$ on Γ $\Longrightarrow$

$$\int_{\Gamma^\sharp \times \Omega^1} u^0 v \ n_{\Gamma^\sharp} \ ds(x^\sharp) dx^1 = 0.$$

- **Step 2.** Proposition 1 $\Longrightarrow$

$$u^0 = 0 \text{ on } \Gamma^\sharp.$$

■

2.3   Homogenized Model Derivation

Here we provide the *reference proof* of the homogenized model derivation. It uses Proposition 19 as an intermediary result. Let u^ε, the solution of a linear boundary value problem posed in Ω,

$$\begin{cases} -\dfrac{d}{dx}(a^\varepsilon(x) \dfrac{du^\varepsilon(x)}{dx}) = f \text{ in } \Omega \\ u^\varepsilon = 0 \text{ on } \Gamma, \end{cases} \tag{22}$$

where the right-hand side $f \in L^2(\Omega)$, the coefficient $a^\varepsilon \in \mathcal{C}^\infty(\Omega)$ is $\varepsilon\Omega^1$-periodic, and there exist two positive constants α and β independent ε such that

$$0 < \alpha \leq a^\varepsilon(x) \leq \beta. \tag{23}$$

The weak formulation is obtained by multiplication of the differential equation by a test function $v \in \mathcal{C}_0^\infty(\Omega)$ and application of the Green formula,

$$\kappa^0 \int_\Omega a^\varepsilon(x) \frac{du^\varepsilon}{dx} \frac{dv}{dx} \ dx = \kappa^0 \int_\Omega f(x)v(x) \ dx. \tag{24}$$

It is known that its unique solution u^ε is bounded as in (15). Moreover, we assume that for some functions $a^0(x^1)$ and $f^0(x^\sharp)$,

$$T(a^\varepsilon) = a^0 \text{ and } T(f) = f^0(x^\sharp) + O_w(\varepsilon). \tag{25}$$

The next proposition states the homogenized model and is the main result of the *reference proof*. For θ^1 a solution to the microscopic problem (13) with $\mu = 1$, the homogenized coefficient and right-hand side are defined by

$$a^H = \int_{\Omega^1} a^0 \left(1 + \frac{\partial \theta^1}{\partial x^1} \right)^2 dx^1 \text{ and } f^H = \int_{\Omega^1} f^0 \ dx^1. \tag{26}$$

Proposition 24 [Homogenized Model] *The limit u^0 is solution to the weak formulation*

$$\int_{\Omega^\sharp} a^H \frac{du^0}{dx^\sharp} \frac{dv^0}{dx^\sharp} \ dx^\sharp = \int_{\Omega^\sharp} f^H v^0 \ dx^\sharp \tag{27}$$

for all $v^0 \in \mathcal{C}_0^\infty(\Omega^\sharp)$.

The proof is split into three lemmas.

Lemma 25 [Fifth Block: Two-Scale Model] *The couple $(u^0, \widetilde{u}^1)$ is solution to the two-scale weak formulation*

$$\int_{\Omega^\sharp \times \Omega^1} a^0 \left(\frac{\partial u^0}{\partial x^\sharp} + \frac{\partial \widetilde{u}^1}{\partial x^1} \right) \left(\frac{\partial v^0}{\partial x^\sharp} + \frac{\partial v^1}{\partial x^1} \right) \ dx^\sharp dx^1 = \int_{\Omega^\sharp \times \Omega^1} f^0 v^0 \ dx^\sharp dx^1 \tag{28}$$

for any $v^0 \in \mathcal{C}_0^\infty(\Omega^\sharp)$ and $v^1 \in \mathcal{C}_0^\infty(\Omega^\sharp, C_\sharp^\infty(\Omega^1))$.

Proof. We choose the test functions $v^0 \in \mathcal{C}_0^\infty(\Omega^\sharp)$, $v^1 \in \mathcal{C}_0^\infty(\Omega^\sharp, C_\sharp^\infty(\Omega^1))$.

- **Step 1** Posing $v = B(v^0 + \varepsilon v^1)$ in (24) and Proposition 14 $\Longrightarrow$

$$Bv \in \mathcal{C}_0^\infty(\Omega) \text{ and } \kappa^0 \int_\Omega a^\varepsilon \frac{du^\varepsilon}{dx} \frac{dB(v^0 + \varepsilon v^1)}{dx} \ dx = \kappa^0 \int_\Omega f \ B(v^0 + \varepsilon v^1) \ dx.$$

- **Step 2** Propositions 15 and 16 $\Longrightarrow$

$$\kappa^0 \int_\Omega a^\varepsilon \frac{du^\varepsilon}{dx} T^* \left(\frac{\partial v^0}{\partial x^\sharp} + \frac{\partial v^1}{\partial x^1} \right) dx = \kappa^0 \int_\Omega f \ T^*(v^0) dx + O(\varepsilon).$$

- **Step 3** Definition 12 and Proposition 6 $\Longrightarrow$

$$\kappa^1 \int_{\Omega^\sharp \times \Omega^1} T(a^\varepsilon) T(\frac{du^\varepsilon}{dx}) \left(\frac{\partial v^0}{\partial x^\sharp} + \frac{\partial v^1}{\partial x^1} \right) \ dx^\sharp dx^1 = \kappa^1 \int_{\Omega^\sharp \times \Omega^1} T(f) \ v^0 \ dx^\sharp dx^1 + O(\varepsilon). \tag{29}$$

- **Step 4** Definitions (25), Lemma 19, and passing to the limit when $\varepsilon \to 0 \Longrightarrow$

$$\int_{\Omega^\sharp \times \Omega^1} a^0 \left(\frac{\partial u^0}{\partial x^\sharp} + \frac{\partial \widetilde{u}^1}{\partial x^1} \right) \left(\frac{\partial v^0}{\partial x^\sharp} + \frac{\partial v^1}{\partial x^1} \right) \ dx^\sharp dx^1 = \int_{\Omega^\sharp \times \Omega^1} f^0 v^0 \ dx^\sharp dx^1$$

which is the expected result.

■

Lemma 26 [Sixth Block: Microscopic Problem] $\widetilde{u}^1$ *is solution to (13) with* $\mu = \dfrac{\partial u^0}{\partial x^\sharp}$ *and*

$$\frac{\partial \widetilde{u}^1}{\partial x^1} = \frac{\partial u^0}{\partial x^\sharp} \frac{\partial \theta^1}{\partial x^1}.$$

10

Proof. We choose $v^0 = 0$ and $v^1(x^\sharp, x^1) = w(x^1)\varphi(x^\sharp)$ in (28) with $\varphi \in \mathcal{C}^\infty(\Omega^\sharp)$ and $w^1 \in \mathcal{C}^\infty_\sharp(\Omega^1)$.

- **Step 1** Proposition 1, Lemma 20, and the linearity of the integral $\Longrightarrow$

$$\int_{\Omega^1} a^0 \frac{\partial \widetilde{u}^1}{\partial x^1} \frac{\partial w^1}{\partial x^1} \, dx^1 = -\frac{\partial u^0}{\partial x^\sharp} \int_{\Omega^1} a^0 \frac{\partial w^1}{\partial x^1} \, dx^1. \tag{30}$$

- **Step 2** Proposition 18 with $\mu = \dfrac{\partial u^0}{\partial x^\sharp} \Longrightarrow$

$$\frac{\partial \widetilde{u}^1}{\partial x^1} = \frac{\partial u^0}{\partial x^\sharp} \frac{\partial \theta^1}{\partial x^1}$$

as announced.

∎

Lemma 27 *[Seventh Block: Macroscopic Problem]* u^0 *is solution to (27).*

Proof. We choose $v^0 \in \mathcal{C}^\infty_0(\Omega^\sharp)$ and $v^1 = \dfrac{\partial v^0}{\partial x^\sharp} \dfrac{\partial \theta^1}{\partial x^1} \in \mathcal{C}^\infty_0(\Omega^\sharp, C^\infty_\sharp(\Omega^1))$ in (28).

- **Step 1** Lemma 26 $\Longrightarrow$

$$\int_{\Omega^\sharp \times \Omega^1} a^0 \left(\frac{\partial u^0}{\partial x^\sharp} + \frac{\partial \theta^1}{\partial x^1} \frac{\partial u^0}{\partial x^\sharp} \right) \left(\frac{\partial v^0}{\partial x^\sharp} + \frac{\partial \theta^1}{\partial x^1} \frac{\partial v^0}{\partial x^\sharp} \right) \, dx^\sharp dx^1 = \int_{\Omega^\sharp \times \Omega^1} f^0 v^0 \, dx^\sharp dx^1. \tag{31}$$

- **Step 2** Factorizing and definitions (26) $\Longrightarrow$

$$\int_{\Omega^\sharp} a^H \frac{\partial u^0}{\partial x^\sharp} \frac{\partial v^0}{\partial x^\sharp} \, dx^\sharp = \int_{\Omega^\sharp} f^H v^0 \, dx^\sharp.$$

∎

3   Rewriting strategies

In this section we recall the rudiments of rewriting, namely, the definitions of terms over a signature, of substitution and of rewriting rules. We introduce a strategy language: its syntax and semantics in terms of partial functions. This language will allow us to express most of the useful rewriting strategies.

3.1   Term, substitution and rewriting rule.

We start with an example of rewriting rule. We define a set of rewriting variables $\mathcal{X} = \{x, y\}$ and a set of function symbols $\Sigma = \{f, g, a, b, c\}$. A term is a combination of elements of $\mathcal{X} \cup \Sigma$, for instance $f(x)$ or $f(a)$. The rewriting rule $f(x) \rightsquigarrow g(x)$ applied to a term $f(a)$ is a two-step operation. First, it consists in matching the left term $f(x)$ with the input term $f(a)$ by matching the two occurences of the function symbol f, and by matching the rewriting variable x with the function symbol a. Then, the result $g(a)$ of the rewriting operation is obtained by replacing the rewriting variable x occuring in the right hand side $g(x)$ by the subterm a that have been associated to x. In case where a substitution is possible, as in the application of $f(b) \rightarrow g(x)$ to $f(a)$, we say that the rewriting rule fails.

Definition 28 *Let Σ be a countable set of function symbols, each symbol $f \in \Sigma$ is associated with a non-negative integer n, its* arity $ar(f)$ *i.e. the number of arguments of f. Let $\mathcal{X}$ be a countable set of variables such that $\Sigma \cap \mathcal{X} = \emptyset$. The set of terms, denoted by $\mathcal{T}(\Sigma, \mathcal{X})$, is inductively defined by*

11

- $\mathcal{X} \subseteq \mathcal{T}(\Sigma, \mathcal{X})$ *(i.e. every rewriting variable is a term),*
- *for all $f \in \Sigma$ of arity n, and all $t_1, \ldots, t_n \in \mathcal{T}(\Sigma, \mathcal{X})$, the expression $f(t_1, \ldots, t_n) \in \mathcal{T}(\Sigma, \mathcal{X})$ (i.e. the application of function symbols to terms gives rise to terms).*

We denote by Σ_n the subset of Σ of the function symbols of arity n. For instance in the example f and g belong to Σ_1 while a and b belong to Σ_0. Two other common examples of terms are the expressions $Integral(\Omega, f(x), x)$ and $diff(f(x), x)$ which represent the expressions $\int_\Omega f(x)\, dx$ and $\dfrac{df(x)}{dx}$. Notice that $Integral \in \Sigma_3$, $diff \in \Sigma_2$, $f \in \Sigma_1$ and $x, \Omega \in \Sigma_0$. For the sake of simplicity we often keep the symbolic mathematical notation to express the rewriting rules. In the following we see a term as an oriented, ranked and rooted tree as it is usual in symbolic computation. We recall that in a ranked tree the child order is important. For instance the tree associated to the term $Integral(\Omega, f(x), x)$ has $Integral$ as its root which has three children in the order Ω, f, x and f has one child x.

Definition 29 *A* substitution *is a function $\sigma : \mathcal{X} \to \mathcal{T}(\Sigma, \mathcal{X})$ such that $\sigma(x) \neq x$ for $x \in \mathcal{X}$. The set of variables that σ does not map to themselves is called the* domain of σ*, i.e. $Dom(\sigma) = \{x \in \mathcal{X} \mid \sigma(x) \neq x\}$. If $Dom(\sigma) = \{x_1, \cdots, x_n\}$ then we might write σ as $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ for some terms $t_1, .., t_n$. Any substitution σ can be extended to a mapping $\mathcal{T}(\Sigma, \mathcal{X}) \to \mathcal{T}(\Sigma, \mathcal{X})$ as follows: for $x \in \mathcal{X}$, $\hat{\sigma}(x) = \sigma(x)$, and for any non-variable term $s = f(s_1, \cdots, s_n)$, we define $\hat{\sigma}(s) = f(\hat{\sigma}(s_1), \cdots, \hat{\sigma}(s_n))$. To simplify the notation we do not distinguish between a substitution $\sigma : \mathcal{X} \to \mathcal{T}(\Sigma, \mathcal{X})$ and its extension $\hat{\sigma} : \mathcal{T}(\Sigma, \mathcal{X}) \to \mathcal{T}(\Sigma, \mathcal{X})$.*

The application *of a substitution σ to a term t, denoted by $\sigma(t)$, simultaneously replaces all occurrences of variables in t by their σ-images.*

For instance, the maping σ defined by $\sigma(x) = a$ is a substitution and its extension $\hat{\sigma}$ maps $f(x)$ and $g(x)$ into $f(a)$ and $g(a)$.

A *rewriting rule*, is a pair (l, r) where l and r are terms in $\mathcal{T}(\Sigma, \mathcal{X})$; it will also be denoted by $l \rightsquigarrow r$. We observe that for any two terms s, t, there exists at most one substitution σ such that $\sigma(s) = t$. We mention that a rewriting rule stands for the rule application at the top position. It is more useful to be able to apply a rule at arbitrary position, and more generally to specify the way rules are applied. For this purpose we next present a strategy language that allows to built strategies out of basic constructors. To this end, we introduce strategy constructor symbols $;, \rightsquigarrow, \oplus, \mu, etc$ that do not belong to $\Sigma \cup \mathcal{X}$. Informally, the constructor ";" stands for the composition, "$\oplus$" for the left choice, *Some* for the application of a strategy to the immediate subterms of the input term, $\eta(x)$ for the fail as identity constructor, $Child(j, s)$ applies the strategy s to the jth immediate subterm, X is a fixed-point variable, and μ is the fixed-point or the iterator constructor, its purpose is to define recursive strategies. For example, the strategy $\mu X.(s; X)$ stands for $s; s; \ldots$, that is, it is the iteration of the application of s until a fixed-point is reached. The precise semantics of these constructors is given in Definition 31.

Definition 30 *(**Strategy**) Let $\mathcal{F}$ be a finite set of fixed-point variables. A strategy is inductively defined by the following grammar:*

$$s ::= l \rightsquigarrow r \mid s; s \mid s \oplus s \mid \eta(s) \mid Some(s) \mid Child(j, s) \mid X \mid \mu X.s \tag{32}$$

where $j \in \mathbb{N}$ and $X \in \mathcal{F}$. The set of strategies defined from a set of rewriting rules in $\mathcal{T}(\Sigma, \mathcal{X}) \times \mathcal{T}(\Sigma, \mathcal{X})$ is denoted by $\mathcal{S}_\mathcal{T}$.

We denote by $\mathbb{F}$ the failing result of a strategy and $\mathcal{T}^*(\Sigma, \mathcal{X}) = \mathcal{T}(\Sigma, \mathcal{X}) \cup \mathbb{F}$.

Definition 31 *(**Semantics of a strategy**) The semantics of a strategy is a function $[\![.]\!] : \mathcal{S}_{\mathcal{T}(\Sigma, \mathcal{X})} \to (\mathcal{T}^*(\Sigma, \mathcal{X}) \to \mathcal{T}^*(\Sigma, \mathcal{X}))$ defined by its application to each grammar component:*
    $[\![s]\!](\mathbb{F}) = \mathbb{F}$

$$\llbracket l \rightsquigarrow r \rrbracket(t) = \begin{cases} \sigma(r) & \text{if } \sigma(l) = t \\ \mathbb{F} & \text{otherwise} \end{cases}$$

$$\llbracket s_1; s_2 \rrbracket(t) = \llbracket s_2 \rrbracket(\llbracket s_1 \rrbracket(t))$$

$$\llbracket s_1 \oplus s_2 \rrbracket(t) = \begin{cases} \llbracket s_1 \rrbracket(t) & \text{if } \llbracket s_1 \rrbracket(t) \neq \mathbb{F} \\ \llbracket s_2 \rrbracket(t) & \text{otherwise} \end{cases}$$

$$\llbracket \eta(s) \rrbracket(t) = \begin{cases} t & \text{if } \llbracket s \rrbracket(t) = \mathbb{F} \\ \llbracket s \rrbracket(t) & \text{otherwise} \end{cases}$$

$$\llbracket Some(s) \rrbracket(t) = \begin{cases} \mathbb{F} & \text{if } ar(t) = 0 \\ f(\eta(s)(t_1), \ldots, \eta(s)(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \text{ and } \exists i \in [1..n] \text{ s.t. } \llbracket s \rrbracket(t_i) \neq \mathbb{F} \\ \mathbb{F} & \text{otherwise} \end{cases}$$

$$\llbracket Child(j,s) \rrbracket(t) = \begin{cases} \mathbb{F} & \text{if } ar(t) = 0, \text{ or } t = f(t_1, \ldots, t_n) \text{ and } j > n \\ f(t_1, \ldots, t_{j-1}, \llbracket s \rrbracket(t_j), t_{j+1}, \ldots, t_n) & \text{if } t = f(t_1, \ldots, t_n) \text{ and } j \leq n. \end{cases}$$

The semantics of the fixed-point constructor is more subtle. One would write:

$$\llbracket \mu X.s \rrbracket = \llbracket s[X/\mu X.s] \rrbracket \tag{33}$$

but this equation cannot be directly used to define $\llbracket \mu X.s \rrbracket$, since the right-hand side contains as a subphrase the phrase whose denotation we are trying to define. Notice that the equation (33) amounts to saying that $\llbracket \mu X.s \rrbracket$ should be the least fixed-point of the operator F:

$$F(X) = \lambda X^{(\mathcal{T}^*(\Sigma, \mathcal{X}) \to \mathcal{T}^*(\Sigma, \mathcal{X}))} \llbracket s \rrbracket^{(\mathcal{T}^*(\Sigma, \mathcal{X}) \to \mathcal{T}^*(\Sigma, \mathcal{X}))}.$$

Let $D = \mathcal{T}^*(\Sigma, \mathcal{X}) \to \mathcal{T}^*(\Sigma, \mathcal{X})$ and define $\sqsubseteq$ a partial order on D as follows:

$$w \sqsubseteq w' \text{ iff } graph(w) \subseteq graph(w').$$

Let $\perp$ be the function of empty graph, and let

$$F_0 = \perp$$
$$F_n = F(F_{n-1}).$$

One can show, using Knaster-Tarsky fixed-point theorem [Tar55], that F_∞ is the least fixed-point of the operator F, that is

$$F(w) = w \implies F_\infty \sqsubseteq w.$$

Such fixed point equations arises very often in giving denotational semantics to languages with recursive features, for instance the semantics of the loop "while" of the programming languages [SK95, §9, §10].

Example 32 *Out of the basic constructors of strategies given in Definition 30, we built up some useful strategies. The strategy $TopDown(s)$ applies the strategy s to an input term t in a top down way starting from the root, it stops when it succeeds. That is, if the strategy s succeeds on some subterm t' of t, then it is not applied to the proper subterms of t'. The strategy $OuterMost(s)$ behaves exactly like $TopDown(s)$ apart that if the strategy s succeeds on some subterm t' of t, then it is also applied to the proper subterms of t'. The strategy $BottomUp(s)$ (resp. $InnerMost(s)$) behaves like $BottomUp(s)$ (resp. $InnerMost(s)$) but in the opposite direction, i.e. it traverses a term t starting from the leafs. The strategy $Normalizer(s)$ iterates the application of s until a fixed-point is reached. The formal definition of these strategies follows:*

$$TopDown(s) := \mu X.(s \oplus Some(X)),$$
$$OuterMost(s) := \mu X.(s; Some(X)),$$
$$BottomUp(s) := \mu X.(Some(X) \oplus s),$$
$$InnerMost(s) := \mu X.(Some(X); s),$$
$$Normalizer(s) := \mu X.(s; X).$$

Example 33 *Let the variable set $\mathcal{X} = \{y, z, t, w\}$ and the partition $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$ of the set of function symbols with respect to their arity with $\Sigma_0 = \{x, x^1, x^2, \partial\Omega, \Omega, \varepsilon\}$, $\Sigma_1 = \{u, v, n, O, B\}$, $\Sigma_2 = \{derivative\}$, $\Sigma_3 = \{Integral\}$ with obvious definitions. We present the strategy that rewrites the expression*

$$\Psi = \int_{\partial\Omega} u(x) \ n(x) \ B(v(x^1, x^2)) \ dx - \int_{\Omega} u(x) \ \frac{d}{dx}(B(v(x^1, x^2))) \ dx + O(\varepsilon),$$

taking into account that $B(v)$ vanishes on the boundary $\partial\Omega$. This term is written under mathematical form for simplicity, but in practice it is written from the above defined symbol of functions. Remark that the expression $B(v(x^1, x^2))$ is a function of the variable x but this does not appear explicitly in this formulation. Such a case cannot appear when the grammar for terms introduced in the next section is used. We need the two rewriting rules

$$r_1 := \int_{\partial\Omega} w \ dt \rightsquigarrow \int_{\partial\Omega} w \ dt,$$
$$r_2 := B(v(z, y)) \rightsquigarrow 0,$$

and the strategy $TopDown$ already defined. Notice that the rule r_1 has not effect but to detect the presence of the integral over the boundary. Finally, the desired strategy is:

$$F := TopDown(r_1; TopDown(r_2)),$$

and the result is

$$[\![F]\!](\Psi) = \int_{\partial\Omega} u(x) \ n(x) \ B(0) \ dx - \int_{\Omega} u(x) \ \frac{d}{dx}(B(v(x^1, x^2))) \ dx + O(\varepsilon).$$

3.2 Rewriting modulo equational theories

So far the semantics of strategies does not take into account the properties of some function symbols, e.g. associativity and commutativity equalities of "+". In particular the application of the rule $a + b \rightsquigarrow f(a, b)$ to the term $(a + c) + b$ fails. More generally we next consider the rewriting modulo an equational theory, i.e. a theory that is axiomatized by a set of equalities.

For the sake of illustration, we consider the commutativity and associativity theory of $+$, $E = \{x + y = y + x, (x + y) + z = x + (y + z)\}$ and the rewrite rule $f(x + y) \rightsquigarrow f(x) + f(y)$ applying the linearity rule of a function f. Its application to the term $f((a + b) + c)$ modulo E yields the set of terms $\{f(a + b) + f(c), f(a) + f(b + c), f(b) + f(a + c)\}$. In the following, we define part of the semantics of a strategy modulo a theory, we use the notation $\mathcal{P}(\mathcal{T}(\Sigma, \mathcal{X}))$ to denote the set of subsets of $\mathcal{T}(\Sigma, \mathcal{X})$.

Definition 34 *(**Semantics of a strategy modulo**) Let be E be a finitary equational theory, the semantics of a strategy modulo E is a function $[\![.]\!]^E : \mathcal{S}_{\mathcal{T}(\Sigma, \mathcal{X})} \to (\mathcal{P}(\mathcal{T}^*(\Sigma, \mathcal{X})) \to \mathcal{P}(\mathcal{T}^*(\Sigma, \mathcal{X})))$ that is partly defined by*

$$[\![s]\!]^E(\{t_1, \ldots, t_n\}) = \cup_{i=1}^n [\![s]\!]^E(t_i)$$
$$[\![l \rightsquigarrow r]\!]^E(t_1) = \cup_j \{\sigma_j(r)\} \ if \ E \implies \sigma_j(l) = t,$$
$$[\![s_1; s_2]\!]^E(t) = [\![s_2]\!]^E([\![s_1]\!]^E(t))$$
$$[\![s_1 \oplus s_2]\!]^E(t) = \begin{cases} [\![s_1]\!]^E(t) & if \ [\![s_1]\!](t) \neq \{\mathbb{F}\} \\ [\![s_2]\!]^E(t) & otherwise \end{cases}$$
$$[\![\eta(s)]\!]^E(t) = \begin{cases} \{t\} & if \ [\![s]\!]^E(t) = \{\mathbb{F}\} \\ [\![s]\!]^E(t) & otherwise. \end{cases}$$

The semantics of *Some* and *Child* is more complex and we do not detail it here. The semantics of the fixed-point operator is similar to the one given in the rewriting modulo an empty theory.

3.3 Conditional rewriting

Rewriting with conditional rules, also known as conditional rewriting, extends the basic rewriting with the notion of condition. A conditional rewrite rule is a triplet:

$$(l, r, c)$$

where c is a constraint expressed in some logic. The semantics of the rule application is given by

$$[\![(l, r, c)]\!]^E(t) = \begin{cases} \cup_j \{\sigma_j(r)\} & \text{if the formula } \sigma_j(c) \text{ can be derived from } E, \\ \mathbb{F} & \text{otherwise.} \end{cases}$$

The set of strategies defined over rewriting rules $(l, r, c) \in \mathcal{T} \times \mathcal{T} \times \mathcal{T}_c$ is denoted by $\mathcal{S}_{\mathcal{T}, \mathcal{T}_c}$.

3.4 Rewriting with memory

Some definitions or computations require storing the history of the transformations of some terms. To carry on, we introduce a particular function symbol $\mathbb{M} \in \Sigma_2$ of arity two to represent the memory. Intuitively the term $\mathbb{M}(t_1, t_2)$ represents the term t_1, besides the additional information that t_2 was transformed to t_1 at an early stage. From this consideration if follows that any strategy applied to $\mathbb{M}(t_1, t_2)$ should only be applied to t_1. Formally, we define the semantics of strategy application taking into account the memory as a partial function: $[\![.]\!]_{\mathbb{M}} : \mathcal{S}_{\mathcal{T}(\Sigma, \mathcal{X})} \to (\mathcal{T}^*(\Sigma, \mathcal{X}) \to \mathcal{T}^*(\Sigma, \mathcal{X}))$ so that:

$[\![s]\!]_{\mathbb{M}}(t) = \mathbb{M}([\![s]\!]_{\mathbb{M}}(t_1), t_2)$ if $t = \mathbb{M}(t_1, t_2)$, and behaves like $[\![.]\!]$, otherwise. That is,

$[\![s]\!]_{\mathbb{M}}(\mathbb{F}) = \mathbb{F}$

$$[\![l \rightsquigarrow r]\!]_{\mathbb{M}}(t) = \begin{cases} \sigma(r) & \text{if } \sigma(l) = t \\ \mathbb{F} & \text{otherwise} \end{cases}$$

$[\![s_1; s_2]\!]_{\mathbb{M}}(t) = [\![s_2]\!]_{\mathbb{M}}([\![s_1]\!]_{\mathbb{M}}(t))$

$$[\![s_1 \oplus s_2]\!]_{\mathbb{M}}(t) = \begin{cases} [\![s_1]\!]_{\mathbb{M}}(t) & \text{if } [\![s_1]\!]_{\mathbb{M}}(t) \neq \mathbb{F} \\ [\![s_2]\!]_{\mathbb{M}}(t) & \text{otherwise} \end{cases}$$

etc.

4 A Symbolic Computation Framework for Model Derivation

In this section we propose a framework for the two-scale model proofs. As in Example 33, the latter are formulated as rewriting strategies. We notice that the following framework differs from that used in Example 33 in that it allows for the complete representation of the data. It does not rely on external structures such as hash tables. To this end, we define the syntax of the mathematical expressions by means of a grammar $\mathcal{G}$.

4.1 A Grammar for Mathematical Expressions

The grammar includes four rules to built terms for mathematical functions $\mathcal{F}$, regions $\mathcal{R}$, mathematical variables $\mathcal{V}$, and boundary conditions $\mathcal{C}$. It involves Σ_{Reg}, Σ_{Var}, Σ_{Fun}, Σ_{Oper}, and Σ_{Cons} which are sets of names of regions, variables, functions, operators, and constants so subsets of Σ_0. Empty expressions in Σ_{Reg} and Σ_{Fun} are denoted by $\perp_{\mathcal{R}}$ and $\perp_{\mathcal{F}}$. The set of usual algebraic operations $\Sigma_{Op} = \{+, -, \times, /, \hat{}\}$ is a subset of Σ_2. The elements of $\Sigma_{Type} = \{Unknown, Test, Known, \perp_{Type}\} \subset \Sigma_0$, $\perp_{Type}$ denoting the empty expression, are to specify the nature of a function, namely an unknown function (as u^ε, u^0, u^1 in the proof), a test function (as v, v^0, v^1) in a weak formulation or another known function (as a^ε, f^ε, a^0, f^0 or n_{Γ^1}). The

boundary conditions satisfied by a function are specified by the elements of $\Sigma_{BC} = \{d, n, pd, apd, t\} \subset \Sigma_0$ to express that it satisfies Dirichlet, Neuman, periodic, anti-periodic or transmission conditions. The grammar also involve the symbols of functions Reg, Fun, IndexedFun, IndexedReg, IndexedVar, Oper, Var, and BC that define regions, mathematical functions, indexed functions or regions or variables, operators, mathematical variables and boundary conditions. The grammar reads as

$$
\begin{aligned}
\mathcal{F} ::= \; & \circledast \, (\mathcal{F}, \mathcal{F}) \; | \; d \; | \; \mathcal{V} \; | \\
& \texttt{Fun}(f, [\mathcal{V}, \ldots, \mathcal{V}], [\mathcal{C}, \ldots, \mathcal{C}], K) \; | \\
& \texttt{IndexedFun}(\mathcal{F}, \mathcal{V}) \; | \\
& \texttt{Oper}(A, [\mathcal{F}, \ldots, \mathcal{F}], [\mathcal{V}, \ldots, \mathcal{V}], [\mathcal{V}, \ldots, \mathcal{V}], [d, \ldots, d]) \; | \\
& \perp_{\mathcal{F}} \; | \; \mathbb{M}(\mathcal{F}, \mathcal{F}), \\
\mathcal{R} ::= \; & \texttt{Reg}(\Omega, [d, \ldots, d], \{\mathcal{R}, \ldots, \mathcal{R}\}, \mathcal{R}, \mathcal{F}) \; | \\
& \texttt{IndexedReg}(\mathcal{F}, \mathcal{V}) \; | \\
& \perp_{\mathcal{R}} \; | \; \mathbb{M}(\mathcal{R}, \mathcal{R}), \\
\mathcal{V} ::= \; & \texttt{Var}(x, \mathcal{R}) \; | \; \texttt{IndexedVar}(\mathcal{V}, \mathcal{V}) \; | \; \mathbb{M}(\mathcal{V}, \mathcal{V}), \\
\mathcal{C} ::= \; & \texttt{BC}(c, \mathcal{R}, \mathcal{F}) \; | \; \mathbb{M}(\mathcal{C}, \mathcal{C}),
\end{aligned}
$$

where the symbols Ω, d, $\circledast$, f, K, A, x and c hold for any function symbols in Σ_{Reg}, Σ_{Cons}, Σ_{Op}, Σ_{Fun}, Σ_{Type}, Σ_{Oper}, Σ_{Var}, and Σ_{BC}. The arguments of a region term are its region name, the list of its space directions (e.g. [1,3] for a plane in the variables (x_1, x_3)), the (possibly empty) set of subregions, the boundary and the outward unit normal. Those of a function term are its function name, the list of the mathematical variables that range over its domain, its list of boundary conditions, and its nature. Those for an indexed region or variable or function term are its function or variable term and its index (which should be discrete). For an operator term these are its name, the list of its arguments, the list of mathematical variable terms that it depends, the list of mathematical variable terms of its co-domain (useful e.g. for T when the image cannot be deduced from the initial set), and a list of parameters. Finally, the arguments of a boundary condition term are its type, the boundary where it applies and an imposed function if there is one. For example, the imposed function is set to 0 for an homogeneous Dirichlet condition and there is no imposed function in a periodicity condition. We shall denote by $\mathcal{T}_{\mathcal{R}}(\Sigma, \emptyset)$, $\mathcal{T}_{\mathcal{F}}(\Sigma, \emptyset)$, $\mathcal{T}_{\mathcal{V}}(\Sigma, \emptyset)$, and $\mathcal{T}_{\mathcal{C}}(\Sigma, \emptyset)$ the set of terms generated by the grammar starting from the non-terminal $\mathcal{R}$, $\mathcal{F}$, $\mathcal{V}$, and $\mathcal{C}$. The set of all terms generated by the grammar (i.e. starting from $\mathcal{R}$, $\mathcal{F}$, $\mathcal{V}$, or $\mathcal{C}$) is denoted by $\mathcal{T}_{\mathcal{G}}(\Sigma, \emptyset)$. Finally, we also define the set of terms $\mathcal{T}_{\mathcal{G}}(\Sigma, \mathcal{X})$ where each non-terminal $\mathcal{R}$, $\mathcal{F}$, $\mathcal{V}$, and $\mathcal{C}$ can be replaced by a rewriting variable in $\mathcal{X}$. Equivalently, it can be generated by the extension of $\mathcal{G}$ obtained by adding " | x" with $x \in \mathcal{X}$ in the definition of each non-terminal term. Or, by adding $N ::= x$, with $x \in \mathcal{X}$ for each non-terminal N.

Example 35 *Throughout this paper, an underlined symbol represents a shortcut whose name corresponds to the term name. For instance,*

$$
\underline{\Omega} = \texttt{Reg}(\Omega, [2], \emptyset, \underline{\Gamma}, \underline{n}), \; \text{where } \underline{\Gamma} = \texttt{Reg}(\Gamma, [], \emptyset, \perp_{\mathcal{R}}, \perp_{\mathcal{F}}),
$$
$$
\underline{n} = \texttt{Fun}(n, [\underline{x'}], [], Known), \; \underline{x'} = \texttt{Var}(x, \underline{\Omega'}) \; \text{and } \underline{\Omega'} = \texttt{Reg}(\Omega, [2], \emptyset, \underline{\Gamma}, \perp_{\mathcal{F}})
$$

represents a region-term a one-dimensional domain named Ω, oriented in the direction x_2, with boundary $\underline{\Gamma}$ and with outward unit normal $\underline{n}$. The shortcut $\underline{\Gamma}$ is also for a region term representing the boundary named Γ. As it can be understood from this example, except names all other fields can be void terms or empty lists.

Example 36 *An unknown function $u(x)$ defined on $\underline{\Omega}$ satisfying homogeneous Dirichlet boundary condition $u(x) = 0$ on $\underline{\Gamma}$ is represented by the function-term,*

$$
\underline{u(x)} = \texttt{Fun}(u, [\underline{x}], \texttt{Cond}(d, \underline{\Gamma}, 0), \texttt{Unknown}) \; \text{where} \quad \underline{x} = \texttt{Var}(x, \underline{\Omega}).
$$

16

4.2   Short-cut Terms

For the sake of conciseness, we introduce shortcut terms that are constantly used in the end of the paper: $\underline{\Omega} \in \mathcal{T}_\mathcal{R}(\Sigma, \mathcal{X})$, $\underline{x} \in \mathcal{T}_\mathcal{V}(\Sigma, \mathcal{X})$ defined in $\underline{\Omega}$, $\underline{I} \in \mathcal{T}_\mathcal{R}(\Sigma, \mathcal{X})$ used for (discrete) indices, $\underline{i} \in \mathcal{T}_\mathcal{V}(\Sigma, \mathcal{X})$ used as an index defined in $\underline{I}$, $\underline{u} \in \mathcal{T}_\mathcal{F}(\Sigma, \mathcal{X})$ or $\underline{u}(\underline{x}) \in \mathcal{T}_\mathcal{F}(\Sigma, \mathcal{X})$ to express that it depends on the variable $\underline{x}$ and $\underline{u}_i$ the indexed-term of the function $\underline{u}$ indexed by $\underline{i}$. Similar definitions can be given for the other notations used in the proof as $\underline{\Omega}^\sharp$, $\underline{x}^\sharp$, $\underline{\Omega}^1$, $\underline{x}^1$, $\underline{\Omega}'$, $\underline{x}'$, $\underline{v}(\underline{x}^\sharp, \underline{x}^1)$ etc. The operators necessary for the proof are the integral, the derivative, the two-scale transform T, its adjoint T^*, and B. In addition, for some extensions of the reference proof we shall use the discrete sum.

Instead of writing operator-terms as defined in the grammar, we prefer to use the usual mathematical expressions. The table below establishes the correspondance between the two formulations.

$$\int \underline{u}\, d\underline{x} \equiv \mathtt{Oper}(\mathtt{Integral}, \underline{u}, [\underline{x}], [], []),$$

$$\frac{\partial \underline{u}}{\partial \underline{x}} \equiv \mathtt{Oper}(\mathtt{Partial}, \underline{u}, [\underline{x}], [\underline{x}], []),$$

$$tr(\underline{u}, \underline{x})(\underline{x}') \equiv \mathtt{Oper}(\mathtt{Restriction}, \underline{u}, [\underline{x}], [\underline{x}'], []),$$

$$T(\underline{u}, \underline{x})(\underline{x}^\sharp, \underline{x}^1) \equiv \mathtt{Oper}(T, \underline{u}, [\underline{x}], [\underline{x}^\sharp, \underline{x}^1], [\varepsilon]),$$

$$T^*(\underline{v}, [\underline{x}^\sharp, \underline{x}^1])(\underline{x}) \equiv \mathtt{Oper}(T^*, \underline{v}, [\underline{x}^\sharp, \underline{x}^1], [\underline{x}], [\varepsilon]),$$

$$B(\underline{v}, [\underline{x}^\sharp, \underline{x}^1])(\underline{x}) \equiv \mathtt{Oper}(B, \underline{v}, [\underline{x}^\sharp, \underline{x}^1], [\underline{x}], [\varepsilon]),$$

$$\sum_{\underline{i}} \underline{u}_i \equiv \mathtt{Oper}(\mathtt{Sum}, \underline{u}_i, [\underline{i}], [], []).$$

The multiplication and exponentiation involving two terms f and g are written fg and f^g as usual in mathematics. All these conventions have been introduced for terms in $\mathcal{T}(\Sigma, \emptyset)$. For terms in $\mathcal{T}(\Sigma, X)$ as those encoutered in rewriting rules, the rewriting variables can replace any of the above short cut terms.

Example 37 *The rewriting rule associated to the Green rule (12) reads*

$$\int \frac{\partial u}{\partial \underline{x}} v\, d\underline{x} \rightsquigarrow -\int u \frac{\partial v}{\partial \underline{x}}\, d\underline{x} + \int tr(u)\, tr(v)\, n\, d\underline{x}'.$$

with the short-cuts $\underline{\Gamma} = \mathtt{Reg}(\Gamma, d1, \emptyset, \perp_\mathcal{R}, \perp_\mathcal{F})$, $\underline{\Omega} = \mathtt{Reg}(\Omega, d2, \emptyset, \underline{\Gamma}, n)$, $\underline{x} = \mathtt{Var}(x, \underline{\Omega})$ *and* $\underline{x}' = \mathtt{Var}(x, \underline{\Gamma})$. *The other symbols u, v, x, Ω, Γ, $d1$, $d2$, n are rewriting variables, and for instance*

$$\frac{\partial u}{\partial x} \equiv \mathtt{Oper}(\mathtt{Partial}, u, x, [], []).$$

Applying this rule according to an appropriate strategy, say the top down strategy, to a term in $\mathcal{T}(\Sigma, \emptyset)$ like

$$\Psi = \int \frac{\partial \underline{f}(\underline{z})}{\partial \underline{z}} \underline{g}(\underline{z})\, d\underline{z},$$

for a given variable term $\underline{z}$ and function terms $\underline{f}$, $\underline{g}$. As expected, the result is

$$-\int \underline{f}\, \frac{\partial \underline{g}}{\partial \underline{z}}\, d\underline{z} + \int \underline{f}\, \underline{g}\, \underline{n}\, d\underline{z}'$$

with evident notations for $\underline{n}$ and $\underline{z}'$.

4.3 A Variable Dependency Analyzer

The *variable dependency analyzer* Θ is related to *effect systems* in computer science [MM09]. It is a function from $\mathcal{T}_{\mathcal{F}}(\Sigma, \emptyset)$ to the set $\mathcal{P}(\mathcal{T}_{\mathcal{V}}(\Sigma, \emptyset))$ of the parts of $\mathcal{T}_{\mathcal{V}}(\Sigma, \emptyset)$. When applied to a term $t \in \mathcal{T}_{\mathcal{F}}(\Sigma, \emptyset)$, it returns the set of mathematical variables on which t depends. The analyzer Θ is used in the condition part of some rewriting rules and is inductively defined by

$$\Theta(\underline{d}) = \emptyset \text{ for } \underline{d} \in \Sigma_{Cons},$$
$$\Theta(\underline{x}) = \{\underline{x}\} \text{ for } \underline{x} \in \mathcal{T}_{\mathcal{V}}(\Sigma, \emptyset),$$
$$\Theta(\circledast(\underline{u}, \underline{v})) = \Theta(\underline{u}) \cup \Theta(\underline{v}) \text{ for } \underline{u}, \underline{v} \in \mathcal{T}_{\mathcal{F}}(\Sigma, \emptyset) \text{ and } \circledast \in \Sigma_{Op},$$
$$\Theta(\perp_{\mathcal{F}}) = \emptyset,$$
$$\Theta(\underline{u}(\underline{x}^1, .., \underline{x}^n)) = \{\underline{x}^1, .., \underline{x}^n\} \text{ for } \underline{u} \in \mathcal{T}_{\mathcal{F}}(\Sigma, \emptyset) \text{ and } \underline{x}^1, .., \underline{x}^n \in \mathcal{T}_{\mathcal{V}}(\Sigma, \emptyset),$$
$$\Theta(\underline{u}_{\underline{i}}) = \Theta(\underline{u}) \text{ for } \underline{u} \in \mathcal{T}_{\mathcal{V}}(\Sigma, \emptyset) \text{ and } \underline{i} \in \mathcal{T}_{\mathcal{V}}(\Sigma, \emptyset),$$
$$\Theta([\underline{u}^1, \ldots, \underline{u}^n]) = \Theta(\underline{u}^1) \cup \cdots \cup \Theta(\underline{u}^n) \text{ for } \underline{u}^1, \ldots, \underline{u}^n \in \mathcal{T}_{\mathcal{F}}(\Sigma, \emptyset).$$

The definition of Θ on the operator-terms is done case by case,

$$\Theta(\int \underline{u}\, d\underline{x}) = \Theta(\underline{u}) \setminus \Theta(\underline{x}),$$
$$\Theta(\frac{\partial \underline{u}}{\partial \underline{x}}) = \begin{cases} \Theta(\underline{u}) \text{ if } \Theta(\underline{x}) \subseteq \Theta(u), \\ \emptyset \text{ otherwise,} \end{cases}$$
$$\Theta(tr(\underline{u}, \underline{x})(\underline{x}')) = \Theta(\underline{x}'),$$
$$\Theta(T(\underline{u}, \underline{x})(\underline{x}^\sharp, \underline{x}^1)) = (\Theta(\underline{u}) \setminus \Theta(\underline{x})) \cup \Theta([\underline{x}^\sharp, \underline{x}^1]) \text{ if } \Theta(\underline{x}) \cap \Theta(\underline{u}) \neq \emptyset,$$
$$\Theta(T^*(\underline{v}, [\underline{x}^\sharp, \underline{x}^1])(\underline{x})) = (\Theta(\underline{v}) \setminus \Theta([\underline{x}^\sharp, \underline{x}^1])) \cup \Theta(\underline{x}) \text{ if } \Theta([\underline{x}^\sharp, \underline{x}^1]) \cap \Theta(\underline{v}) \neq \emptyset,$$
$$\Theta(B(\underline{v}, [\underline{x}^\sharp, \underline{x}^1])(\underline{x}))) = (\Theta(\underline{v}) \setminus \Theta([\underline{x}^\sharp, \underline{x}^1])) \cup \Theta(\underline{x}) \text{ if } \Theta([\underline{x}^\sharp, \underline{x}^1]) \cap \Theta(\underline{v}) \neq \emptyset,$$
$$\Theta(\sum_{\underline{i}} \underline{u}_{\underline{i}}) = \bigcup_{\underline{i}} \Theta(\underline{u}_{\underline{i}}).$$

We observe that these definitions are not very general, but they are sufficient for the applications of this paper. To complete the definition of Θ, it remains to define it on memory terms,

$$\Theta(\mathbb{M}(\underline{u}, \underline{v})) = \Theta(\underline{u}).$$

Example 38 *For*
$$\Psi = \int_{\underline{\Omega}^\sharp} [\int_{\underline{\Omega}^1} T(\underline{u}(\underline{x}), \underline{x})(\underline{x}^\sharp, \underline{x}^1) \frac{\partial \underline{v}(\underline{x}^\sharp, \underline{x}^1)}{\partial \underline{x}^1} d\underline{x}^1] d\underline{x}^\sharp \in \mathcal{T}_{\mathcal{F}}(\Sigma, \emptyset),$$

the set $\Theta(\Psi)$ of mathematical variables on which Ψ depends is hence inductively computed as follows:
$\Theta(\underline{u}(\underline{x})) = \{\underline{x}\}$, $\Theta(T(\underline{u}(\underline{x}), \underline{x})(\underline{x}^\sharp, \underline{x}^1)) = \{\underline{x}^\sharp, \underline{x}^1\}$, $\Theta(\underline{v}(\underline{x}^\sharp, \underline{x}^1)) = \{\underline{x}^\sharp, \underline{x}^1\}$, $\Theta(\frac{\partial \underline{v}(\underline{x}^\sharp, \underline{x}^1)}{\partial \underline{x}^1}) = \{\underline{x}^\sharp, \underline{x}^1\}$, $\Theta(T(\underline{u}(\underline{x}), \underline{x})$ $(\underline{x}^\sharp, \underline{x}^1) \frac{\partial \underline{v}(\underline{x}^\sharp, \underline{x}^1)}{\partial \underline{x}^1}) = \{\underline{x}^\sharp, \underline{x}^1\}$, $\Theta(\int_{\underline{\Omega}^1} T(\underline{u}(\underline{x}), \underline{x})(\underline{x}^\sharp, \underline{x}^1) \frac{\partial \underline{v}(\underline{x}^\sharp, \underline{x}^1)}{\partial \underline{x}^1} d\underline{x}^1) = \{\underline{x}^\sharp\}$, *and* $\Theta(\Psi) = \emptyset$, *that is,* Ψ *is a constant function.*

4.4 Formulation of the Symbolic Framework for Model Derivation

Now we are ready to define the framework for two-scale model derivation by rewriting. To do so, the rewriting rules are restricted to left and right terms $(l, r) \in \mathcal{T}_{\mathcal{G}}(\Sigma, \mathcal{X}) \times \mathcal{T}_{\mathcal{G}}(\Sigma, \mathcal{X})$. Their conditions c are formulas generated by a grammar, not explicited here, combining terms in $\mathcal{T}_{\mathcal{G}}(\Sigma, \mathcal{X})$ with the usual logical operators in $\Lambda = \{\vee, \wedge,], \in\}$. It also involves operations with the dependency analyzer Θ. The set of terms generated by this grammar is denoted by $\mathcal{T}_{\mathcal{L}}(\Sigma, \mathcal{X}, \mathcal{G}, \Theta, \Lambda)$.

It remains to argue that, given a strategy s in $\mathcal{S}_{\mathcal{T}_\mathcal{G}(\Sigma,\mathcal{X}),\mathcal{T}_\mathcal{L}(\Sigma,\mathcal{X},\mathcal{G},\Theta,\Lambda)}$, the set of terms $\mathcal{T}_\mathcal{G}(\Sigma,\emptyset)$ is closed under the application of s. It is sufficient to show that for each rewriting r rule in s, the application of r to any term $t \in \mathcal{T}_\mathcal{G}(\Sigma,\emptyset)$ at any position yields a term in $\mathcal{T}_\mathcal{G}(\Sigma,\emptyset)$. As an example, $\mathcal{T}_\mathcal{G}(\Sigma,\emptyset)$ is not closed under the application of the rule $x \rightsquigarrow \underline{\Omega}$, where x is a variable. But it is closed under the application of the linearity rule $\int_z f + g\,dx \rightsquigarrow \int_z f\,dx + \int_z g\,dx$ at any position, where f, g, x, z are rewriting variables. The argument is, since $\int_z f + g\,dx \in \mathcal{T}_\mathcal{F}(\Sigma,\emptyset)$, then $f + g \in \mathcal{T}_\mathcal{F}(\Sigma,\emptyset)$, and hence $f, g \in \mathcal{T}_\mathcal{F}(\Sigma,\emptyset)$. Thus, $\int_z f\,dx + \int_z g\,dx \in \mathcal{T}_\mathcal{F}(\Sigma,\emptyset)$. That is, a term in $\mathcal{T}_\mathcal{F}(\Sigma,\emptyset)$ is replaced by a another term in $\mathcal{T}_\mathcal{F}(\Sigma,\emptyset)$. A more general setting that deals with the closure of regular languages under specific rewriting strategies can be found in [GGJ09].

A model derivation is divided into several intermediary lemmas. Each of them is intended to produce a new property that can be expressed as one or few rewriting rules to be applied in another part of the derivation. Since dynamical creation of rules is not allowed, a strategy is covering one lemma only and is operating with a fixed set of rewriting rules. The conversion of a result of a strategy to a new set of rewriting rules is done by an elementary external operation that is not a limitation for generalizations of proofs. The following definition summarizes the framework of symbolic computation developed in this paper.

Definition 39 *The components of the quintuplet* $\Xi = \langle \Sigma, \mathcal{X}, E, \mathcal{G}, \Theta \rangle$ *provide a framework for symbolic computation to derive multi-scale models. A two-scale model derivation is expressed as a strategy* $\pi \in \mathcal{S}_{\mathcal{T}_\mathcal{G}(\Sigma,\mathcal{X}),\mathcal{T}_\mathcal{L}(\Sigma,\mathcal{X},\mathcal{G},\Theta,\Lambda)}$ *for which the semantics* $[\![\pi]\!]^E$ *is applicable to an initial expression* $\Psi \in \mathcal{T}(\Sigma,\emptyset)$.

In the end of this section we argue that this framework is in the same time relatively simple, it covers the *reference model* derivation and it allows for the extensions presented in the next section.

The grammar of terms is designed to cover all mathematical expressions occuring in the proof of the *reference model* as well as of their generalizations. A term produced by the grammar includes locally all useful information. This avoids the use of external tables and facilitates design of rewriting rules, in particular to take into account the context of subterms to be transformed. It allows also for local definitions, for instance a same name of variable x can be used in different parts of a same term with different meaning, which is useful for instance in integrals. A limitation regarding generalizations presented in the next section, is that the grammar must cover by anticipation all needed features. This drawback should be fixed in another work by supporting generalization of grammars in the same time as generalization of proofs.

Each step in the proof consists in replacing parts of an expression according to a known mathematical property. This is well done, possibly recursively, using rewriting rules together with strategies allowing for precise localization. Some steps need simplifications and often use the second linearity rule of a linear operator, $A(\lambda u) = \lambda A u$ when λ is a scalar (or is independent of the variables in the initial set of A). So variable dependency of each subterm should be determined, this is precisely what Θ, the *variable dependency analyzer*, is producing. The other simplifications do not require the use of Θ. In addition to the grammar $\mathcal{G}$, the analyzer Θ must be upgraded in view of each new extension.

In all symbolic computation based on the grammar $\mathcal{G}$, it is implicitly assumed that the derivatives, the integrals and the traces (i.e. restriction of a function to the boundary) are well defined since the regularity of functions is not encoded.

Due to the algebraic nature of the mathematical proofs, this framework has been formulated by considering these proofs as a calculus rather than formal proofs that can be formalized and checked with a proof assistant [BC04, Won96]. Indeed, this is far simpler and allows, from a very small set of tools, for building significant mathematical derivation. To cover broader proofs, the framework must be changed by extending the grammar and the variable dependency analyzer only. Yet, the language Tom [BBK$^+$07] does not provide a complete environment for the implementation of our framework since it does not support the transformation of rewriting rules, despite it provides a rich strategy language and a module for the specification of the grammar.

5 Transformation of Strategies as Second Order Strategies

For a given rewriting strategy representing a model proof, one would like to transform it to obtain a derivation of more complex models. Transforming a strategy $\pi \in \mathcal{S}_{\mathcal{T}(\Sigma, \mathcal{X})}$ is achieved by applying strategies to the strategy π itself. For this purpose, we consider two levels of strategies: the first order ones $\mathcal{S}_{\mathcal{T}(\Sigma, \mathcal{X})}$ as defined in Definition 30, and the strategies of second order in such a way that second order strategies can be applied to first order ones. That is, the second order strategies are considered as terms in a set $\mathcal{T}(\overline{\Sigma}, \overline{\mathcal{X}})$ of terms where $\overline{\Sigma}$ and $\overline{\mathcal{X}}$ remain to be defined. Given a set of strategies $\mathcal{S}_{\mathcal{T}(\Sigma, \mathcal{X})}$ that comes with a set of fixed-point variables $\mathcal{F}$, we pose $\overline{\Sigma} \supset \Sigma \cup \{\rightsquigarrow, ;, \oplus, Some, Child, \eta, \mu\} \cup \mathcal{F}$. Let $\overline{\mathcal{X}}$ be a set of second order rewriting variables such that $\overline{\mathcal{X}} \cap (\mathcal{X} \cup \overline{\Sigma}) = \emptyset$. Notice that first order rewriting variables and fixed-point variables are considered as constants in $\mathcal{T}(\overline{\Sigma}, \overline{\mathcal{X}})$, i.e. function symbols in $\overline{\Sigma}_0$. Notice also that the arity of the function symbols $\rightsquigarrow, ;, \oplus, Child, \mu$ is two, and the arity of $Some$ and η is one. In particular, the rule $l \rightsquigarrow r$ can be viewed as the term $\rightsquigarrow (l, r)$ with the symbol $\rightsquigarrow$ at the root, and the strategy $\mu X.s$ viewed as the term $\mu(X, s)$. This allows us to define second order strategies $\overline{\mathcal{S}}_{\mathcal{T}(\overline{\Sigma}, \overline{\mathcal{X}})}$ by the grammar

$$\bar{s} ::= l\tilde{\rightsquigarrow}r \mid \overline{\bar{s};\bar{s}} \mid \bar{s}\bar{\oplus}\bar{s} \mid \bar{\eta}(\bar{s}) \mid \overline{Some}(\bar{s}) \mid \overline{Child}(j, \bar{s}) \mid X \mid \bar{\mu}X.\bar{s} \qquad (34)$$

Again we assume that the symbols $\tilde{\rightsquigarrow}, \bar{;}, \bar{\oplus}, \ldots$ of the second order strategies do no belong to $\overline{\Sigma}$. The semantics of the strategies in $\overline{\mathcal{S}}_{\mathcal{T}(\overline{\Sigma}, \overline{\mathcal{X}})}$ are similar to the semantics of first order strategies. In addition, we assume that second order strategies transform first order strategies, to which they are applied, into first order strategies. Composing several second order strategies and applying such composition to a given first order strategy s provide successive transformations of s.
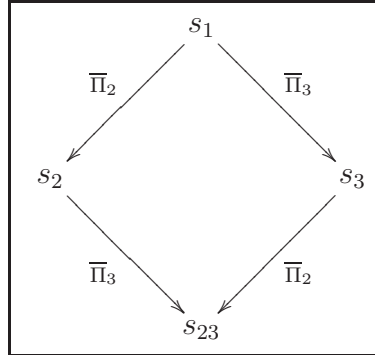

Figure 1: An example of the composition of transformations of strategies.

In the following example we illustrate the extension of an elementary strategy which is a rewriting rule.

Example 40 *For the set $\mathcal{X} = \{i, j, x, x^\sharp, x^1, u, \varepsilon\}$ we define s_1, s_2, s_3, and s_{23} four rewriting rules,*

$$s_1 := T(\frac{\partial u}{\partial x}, x)(x^\sharp, x^1) \rightsquigarrow \frac{1}{\varepsilon}\frac{\partial T(u, x)(x^\sharp, x^1)}{\partial x^1} \text{ for } x \in \Omega \text{ and } (x^\sharp, x^1) \in \Omega^\sharp \times \Omega^1,$$

$$s_2 := T(\frac{\partial u}{\partial x_i}, x)(x^\sharp, x^1) \rightsquigarrow \frac{1}{\varepsilon}\frac{\partial T(u, x)(x^\sharp, x^1)}{\partial x_i^1} \text{ for } x \in \Omega \text{ and } (x^\sharp, x^1) \in \Omega^\sharp \times \Omega^1,$$

$$s_3 := T(\frac{\partial u}{\partial x}, x)(x^\sharp, x^1) \rightsquigarrow \frac{1}{\varepsilon}\frac{\partial T(u, x)(x^\sharp, x^1)}{\partial x^1} \text{ for } x \in \Omega_j \text{ and } (x^\sharp, x^1) \in \Omega_j^\sharp \times \Omega_j^1,$$

$$s_{23} := T(\frac{\partial u}{\partial x_i}, x)(x^\sharp, x^1) \rightsquigarrow \frac{1}{\varepsilon}\frac{\partial T(u, x)(x^\sharp, x^1)}{\partial x_i^1} \text{ for } x \in \Omega_j \text{ and } (x^\sharp, x^1) \in \Omega_j^\sharp \times \Omega_j^1.$$

The rule s_1 is encountered in the reference proof, s_2 is a (trivial) generalization of s_1 in the sense that it applies to multi-dimensional regions Ω^1 referenced by a set of variables $(x_i^1)_i$, and s_3 is a second (trivial) generalization of s_1 on the number of sub-regions $(\Omega_j)_j$, $(\Omega_j^\sharp)_j$ and $(\Omega_j^1)_j$ in Ω, $\Omega^\sharp$ and Ω^1. The rule s_{23} is a generalization combining the two previous generalizations. First, we aim at transforming the strategy s_1 into the strategy s_2 or the strategy s_3. To this end, we introduce two second order strategies with $\overline{\mathcal{X}} = \{v, z\}$ and $\overline{\Sigma} \supset \{i, j, \Omega, \Omega^\sharp, \Omega^1, Partial, IndexedFun, IndexedVar, IndexedReg\}$,

$$\bar{\Pi}_1 := \overline{OuterMost}(\frac{\partial v}{\partial z} \tilde{\leadsto} \frac{\partial v}{\partial z_i})$$

$$\bar{\Pi}_2 := \overline{OuterMost}(\Omega \tilde{\leadsto} \Omega_j); \overline{OuterMost}(\Omega^\sharp \tilde{\leadsto} \Omega_j^\sharp); \overline{OuterMost}(\Omega^1 \tilde{\leadsto} \Omega_j^1)$$

Notice that $\bar{\Pi}_1$ (resp. $\bar{\Pi}_2$) applies the rule $\dfrac{\partial v}{\partial z} \tilde{\leadsto} \dfrac{\partial v}{\partial z_i}$ (resp. $\Omega \tilde{\leadsto} \Omega_j$, $\Omega^\sharp \tilde{\leadsto} \Omega_j^\sharp$, and $\Omega^1 \tilde{\leadsto} \Omega_j^1$) at all of the positions [1] of the input first order strategy so that

$$\bar{\Pi}_1(s_1) = s_2 \ and \ \bar{\Pi}_2(s_1) = s_3.$$

Once $\bar{\Pi}_1$ and $\bar{\Pi}_2$ have been defined, they can be composed to produce s_{23} :

$$\bar{\Pi}_2\bar{\Pi}_1(s_1) = s_{23} \ or \ \bar{\Pi}_1\bar{\Pi}_2(s_1) = s_{23}.$$

The diagram of Figure 1 illustrates the application of $\bar{\Pi}_1$, $\bar{\Pi}_2$ and of their compositions.

The next example shows how an extension can not only change rewriting rules but also to add new ones.

Example 41 *To operate simplifications in the reference model, we use the strategy*

$$s_1 := TopDown(\frac{\partial x}{\partial x} \leadsto 1).$$

In the generalization to multi-dimensional regions, it is replaced by two strategies involving the Kronecker symbol δ, usually defined as $\delta(i, j) = 1$ if $i = j$ and $\delta(i, j) = 0$ otherwise,

$$
\begin{aligned}
s_2 \quad &: \quad = TopDown\left(\frac{\partial x_i}{\partial y_j} \leadsto \delta(i, j), \ x = y\right), \\
s_3 \quad &: \quad = TopDown\left(\delta(i, j) \leadsto 1, \ i = j\right), \\
s_4 \quad &: \quad = TopDown\left(\delta(i, j) \leadsto 0, \ i \neq j\right).
\end{aligned}
$$

The second order strategy that transforms s_1 into the strategy $Normalizer(s_2 \oplus s_3 \oplus s_4)$ is

$$\bar{\Pi} := \overline{TopDown}(s_1 \tilde{\leadsto} s_2 \oplus s_3 \oplus s_4).$$

---

[1]Notice the difference with $\overline{TopDown}$ which could not apply these rules at any position.

6 Implementation and Experiments

The framework presented in Section 4.4 has been implemented in Maple$^\circledR$. The implementation includes, the language *Symbtrans* of strategies already presented in [BGL]. The derivation of the reference model presented in Section 2 has been fully implemented. It starts from an input term which is the weak formulation (24) of the physical problem,

$$\int \underline{a}\frac{\partial \underline{u}}{\partial \underline{x}}\frac{\partial \underline{v}}{\partial \underline{x}} \; d\underline{x} = \int \underline{f} \; \underline{v} \; d\underline{x}, \tag{35}$$

where $\underline{a} = \texttt{Fun}(a, [\underline{\Omega}], [\,], Known)$, $\underline{u} = \texttt{Fun}(u, [\underline{\Omega}], [\underline{Dirichlet}], Unknown)$, $\underline{v} = \texttt{Fun}(u, [\underline{\Omega}], [\underline{Dirichlet}], Test)$, $\underline{\Omega} = \texttt{Reg}(\Omega, [1], \emptyset, \underline{\Gamma}, n_\Omega)$, $\underline{\Gamma} = \texttt{Reg}(\Gamma, [\,], \emptyset, \perp_\mathcal{R}, \perp_\mathcal{F})$, $\underline{Dirichlet} = \texttt{BC}(Dirichlet, \underline{\Gamma}, 0)$ and where the short-cuts of the operators are those of Section 4.2. The information regarding the two-scale transformation is provided through the test functions. For instance, in the first block the proof starts with the expression

$$\Psi = \int \frac{\partial \underline{u}}{\partial \underline{x}} B(\underline{v}(\underline{x}^\sharp, \underline{x}^1)(\underline{x}) \; d\underline{x},$$

where the test function $B(\underline{v}(\underline{x}^\sharp, \underline{x}^1)(\underline{x})$ is also an input, with $\underline{v} = \texttt{Fun}(a, [\underline{x}^\sharp, \underline{x}^1], [\underline{Dirichlet}\sharp], Test)$, $\underline{x}^\sharp = \texttt{Var}(x^\sharp, \underline{\Omega}^\sharp)$, $\underline{x}^1 = \texttt{Var}(x^1, \underline{\Omega}^1)$, $\underline{\Omega}^\sharp = \texttt{Reg}(\Omega^\sharp, [1], \emptyset, \underline{\Gamma}^\sharp, n_{\Omega^\sharp})$, $\underline{\Gamma}^\sharp = \texttt{Reg}(\Gamma^\sharp, [\,], \emptyset, \perp_\mathcal{R}, \perp_\mathcal{F})$, $\underline{\Omega}^1 = \texttt{Reg}(\Omega^1, [1], \emptyset, \underline{\Gamma}^1, n_{\Omega^1})$, $\underline{\Gamma}^1 = \texttt{Reg}(\Gamma^1, [\,], \emptyset, \perp_\mathcal{R}, \perp_\mathcal{F})$, and $\underline{Dirichlet}\sharp = \texttt{BC}(Dirichlet\sharp, \underline{\Gamma}^\sharp, 0)$.

The proof is divided into five strategies corresponding to the five blocks of the proof, each ending by some results transformed into rewriting rules used in the following blocks. The rewriting rules used in the strategies are FO-rules and can be classified into the three categories.

- *Usual mathematical rules:* that represents the properties of the derivation and integration operators, such as the linearity, the chain rule, the Green rule, etc,

- *Specialized rules:* for the properties of the two-scale calculus, as those of the two-scale transform, the approximation of B by the adjoint T^* etc,

- *Auxiliary tools:* for transformations of expressions format that are not related to operator properties such as the rule which transforms $\psi_1 = \psi_2$ into $\psi_1 - \psi_2 = 0$.

	Usual Rules	Specialized Rules	Aux. Tools
Skeleton	53	14	28

Table 1: The number of first order rules used in the reference model.

The Table 1 summarizes the number of first order (FO) rules, used in the reference model, by categories.

The reference model has been extended to cover three different kinds of configurations. To proceed to an extension, the new model derivation is established in a form that is as close as possible of the *reference proof.* The grammar and the dependency analyzer should be completed. Then, the initial data is determined, and second order (SO) strategies yielding the generalized model derivation are found and optimized. As it has been already mentioned, $\mathcal{G}$ and Θ have already been designed to cover the three extensions.

The first generalization is to cover multi-dimensional regions, i.e. $\Omega \subset \mathbb{R}^n$ with $n \geq 1$. When $n = 2$, the initial term is

$$\sum_{\underline{i}=1}^{n}\sum_{\underline{j}=1}^{n}\int \underline{a_{ij}}\frac{\partial \underline{u}}{\partial \underline{x_i}}\frac{\partial \underline{v}}{\partial \underline{x_j}}\ d\underline{x} = \int \underline{f}\ \underline{v}\ d\underline{x},$$

where $\underline{\Omega} = \mathtt{Reg}(\Omega, [1,2], \emptyset, \underline{\Gamma}, n_\Omega), \underline{a_{ij}} = \mathtt{Indexed}(\mathtt{Indexed}(\underline{a}, \underline{j}), \underline{i}), \underline{i} = \mathtt{Var}(i, \underline{I}), \underline{I} = \mathtt{Reg}(I, [1,2], \emptyset, \perp_{\mathcal{R}}, \perp_{\mathcal{F}})$ and the choice of the test function is trivially deduced. Then, the model derivation is very similar to this of the *reference model*, see [LS07], so much so it is obtained simply by applying the SO strategy $\bar{\bar{\Pi}}_1$ defined in Example 40. This extension has been tested on the four first blocks.

The second generalization transforms the *reference model* into a model with several adjacent one-dimensional regions (or intervals) $(\Omega_k)_{k=1,..,m}$ so that Ω is still an interval i.e. $\Omega \subset \mathbb{R}$. For $m = 2$, the initial term is the same as (35) but with $\underline{\Omega} = \mathtt{Reg}(\Omega, [1], \{\underline{\Omega_1}, \underline{\Omega_2}\}, \underline{\Gamma}, n_\Omega), \underline{\Omega_1} = \mathtt{Reg}(\Omega_1, [1], \emptyset, \underline{\Gamma_1}, n_{\Omega_1})$, and $\underline{\Omega_2} = \mathtt{Reg}(\Omega_2, [1], \emptyset, \underline{\Gamma_2}, n_{\Omega_2})$. The two-scale geometries, all variables, all kind of functions and also the operators B and T are defined subregion by subregion. All definitions and properties apply for each subregion, and the proof steps are the same after splitting the integrals over the complete region Ω into integrals over the subregions. The only major change is in the fourth step where the equality $u_1^0 = u_2^0$ at the interface between Ω_1 and Ω_2 which is encoded as transmission conditions in the boundary conditions of u_1^0 and u_2^0.

The third extension transforms the multi-dimensional model obtained from the first generalization to a model related to thin cylindrical regions, in the sense that the dimension of Ω is in the order of ε in some directions $i \in I^\natural$ and of the order 1 in the others $i \in I^\sharp$ e.g. $\Omega = (0,1) \times (0,\varepsilon)$ where $I^\natural = \{2\}$ and $I^\sharp = \{1\}$. The boundary Γ is split in two parts, the lateral part Γ_{lat} and the other parts Γ_{other} where the Dirichlet boundary conditions are replaced by homogeneous Neuman boundary conditions i.e. $\frac{du^\varepsilon}{dx} = 0$. In this special case the integrals of the initial term are over a region which size is of the order of ε so it is required to multiply each side of the equality by the factor $1/\varepsilon$ to work with expressions of the order of 1. Moreover, the macroscopic region differs from Ω, it is equal to $\Omega^\sharp = (0,1)$ when the microscopic region remains unchanged. In general, the definition of the adjoint T^* is unchanged but $(Bv)(x) = v((x_i)_{i\in I^\sharp}, (x - x_c^\sharp)/\varepsilon)$ where $x_c^\sharp$ is the center of the cth cell in $\Omega^\sharp$. It follows that the approximations (10, 11) are between T^* and εB with $\sum_{i\in I^\sharp} x_i^1 \frac{\partial v}{\partial x_i^\sharp}$ instead of $\sum_{i=1}^{n} x_i^1 \frac{\partial v}{\partial x_i^\sharp}$. With these main changes in the definitions and the preliminary properties, the proof steps may be kept unchanged.

	Usual Rules	Specialized Rules	Aux. Tools
Multi-Dimension	6	0	4
Thin-Region	2	0	0
Multi-Region	3	0	0

Table 2: The number of first order rules used in the three extensions.

The mathematical formulation of the second and third extensions has been derived. This allows for the determination of the necessary SO-strategies, but they have not been implemented nor tested. To summarize the results about the principle of extension of strategies, we show its benefit through some statistics. In particular the main concerned is the reusability and the extensibility of existing strategies. The Table 2 shows an estimate of the number of new FO-rules for the three extensions in each category and for the first four blocks.
The Table 3 shows the number of SO-strategies used in each extension. Finally, the Table 4 shows, the ratio of the modified FO-rules and the ratio of the modified FO-strategies. The reusability ratio is high since most of the FO-strategies defined in the skeleton model are reused. Besides very little number of

	Usual Rules	Specialized Rules	Aux. Tools
Multi-Dimension	9	2	3
Thin-Region	0	0	0
Multi-Region	1	0	0

Table 3: The number of second order strategies used in the extension of proofs.

Input model	Resulting model	% Modi. FO-rules	% Modi. FO-strategies
Reference	Multi-Dim.	16.6%	5%
Multi-Dim.	Thin	0	0
Thin	Multi-Reg.	0	2.5%

Table 4: The ratio of modified FO-rules and FO-strategies.

SO-strategies is used in the extensions. This systematic way of the generation of proofs is a promising path that will be further validated within more complex configurations for which the proofs can not obtained by hand. In the future, we plan to introduce dedicated tools to aid in the design of composition of several extensions.

References

[ADH90] Todd Arbogast, Jim Douglas, Jr., and Ulrich Hornung. Derivation of the double porosity model of single phase flow via homogenization theory. *SIAM J. Math. Anal.*, 21:823–836, May 1990.

[BB02] G. Bouchitte and M. Bellieud. Homogenization of a soft elastic material reinforced by fibers. *Asymptotic Analysis*, 32(2):153, 2002.

[BBK+07] Emilie Balland, Paul Brauner, Radu Kopetz, Pierre-Etienne Moreau, and Antoine Reilles. Tom: Piggybacking rewriting on Java. In *the proceedings of the 18th International Conference on Rewriting Techniques and Applications RTA 07*, pages 36–47, 2007.

[BC04] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions.* Texts in Theoretical Computer Science. Springer Verlag, 2004.

[BGL] W. Belkhir, A. Giorgetti, and M. Lenczner. A symbolic transformation language and its application to a multiscale method. Submitted. December 2010, http://arxiv.org/abs/1101.3218v1.

[BKKR01] Peter Borovansky, Claude Kirchner, Hélène Kirchner, and Christophe Ringeissen. Rewriting with strategies in ELAN: a functional semantics. *International Journal of Foundations of Computer Science*, 12(1):69–95, 2001.

[BLM96] Alain Bourgeat, Stephan Luckhaus, and Andro Mikelic. Convergence of the homogenization process for a double-porosity model of immiscible two-phase flow. *SIAM J. Math. Anal.*, 27:1520–1543, November 1996.

[BLP78] A. Bensoussan, J.L. Lions, and G. Papanicolaou. *Asymptotic Methods for Periodic Structures.* North-Holland, 1978.

[BN98] F. Baader and T. Nipkow. *Term rewriting and all that.* Cambridge University Press, 1998.

[CD99] D. Cioranescu and P. Donato. *An introduction to homogenization.* Oxford University Press, 1999.

[CD00] J. Casado-Díaz. Two-scale convergence for nonlinear Dirichlet problems in perforated domains. *Proc. Roy. Soc. Edinburgh Sect. A*, 130(2):249–276, 2000.

[CDG02] D. Cioranescu, A. Damlamian, and G. Griso. Periodic unfolding and homogenization. *C. R. Math. Acad. Sci. Paris*, 335(1):99–104, 2002.

[CDG08] D. Cioranescu, A. Damlamian, and G. Griso. The periodic unfolding method in homogenization. *SIAM Journal on Mathematical Analysis*, 40(4):1585–1620, 2008.

[CFK05] H. Cirstea, G. Faure, and C. Kirchner. A ρ-calculus of explicit constraint application. *Electronic Notes in Theoretical Computer Science*, 117:51 – 67, 2005. Proceedings of the Fifth International Workshop on Rewriting Logic and Its Applications (WRLA 2004).

[CK01] Horatiu Cirstea and Claude Kirchner. The rewriting calculus — Part I and II. *Logic Journal of the Interest Group in Pure and Applied Logics*, 9(3):427–498, May 2001.

[CKLW03] Horatiu Cirstea, Claude Kirchner, Luigi Liquori, and Benjamin Wack. Rewrite strategies in the rewriting calculus. In Bernhard Gramlich and Salvador Lucas, editors, *3rd International Workshop on Reduction Strategies in Rewriting and Programming* , volume 86(4) of *Electronic Notes in Theoretical Computer Science*, pages 18–34, Valencia, Spain, 2003. Elsevier.

[GGJ09] Adrià Gascón, Guillem Godoy, and Florent Jacquemard. Closure of tree automata languages under innermost rewriting. *Electron. Notes Theor. Comput. Sci.*, 237:23–38, April 2009.

[JZKO94] V.V. Jikov, V. Zhikov, M. Kozlov, and O.A. Oleinik. *Homogenization of differential operators and integral functionals.* Springer-Verlag, 1994.

[Len97]   M. Lenczner. Homogénéisation d'un circuit électrique. *C. R. Acad. Sci. Paris Sér. II b*, 324(9):537–542, 1997.

[Len06]   Michel Lenczner. Homogenization of linear spatially periodic electronic circuits. *NHM*, 1(3):467–494, 2006.

[LS07]   M. Lenczner and R. C. Smith. A two-scale model for an array of AFM's cantilever in the static case. *Mathematical and Computer Modelling*, 46(5-6):776–805, 2007.

[MM09]   Daniel Marino and Todd Millstein. A generic type-and-effect system. In *Proceedings of the 4th international workshop on Types in language design and implementation*, TLDI '09, pages 39–50, New York, NY, USA, 2009. ACM.

[SK95]   Kenneth Slonneger and Barry L. Kurtz. *Formal syntax and semantics of programming languages - a laboratory based approach*. Addison-Wesley, 1995.

[Tar55]   Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *The Journal of Symbolic Logic*, 5(4):370, 1955.

[Ter03]   Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theor. Comp. Sci.* Cambridge Univ. Press, 2003.

[Won96]   Wai Wong. A proof checker for hol, 1996.